

AD-751 318

ROFF - A MANUSCRIPT PRINTING PROGRAM:  
USER'S MANUAL

Clifford E. Rhoades, Jr.

Air Force Weapons Laboratory  
Kirtland Air Force Base, New Mexico

November 1972

DISTRIBUTED BY:

**NTIS**

National Technical Information Service  
U. S. DEPARTMENT OF COMMERCE  
5285 Port Royal Road, Springfield Va. 22151

AFWL-TR-72-139

AFWL-TR-  
72-139

AD751318

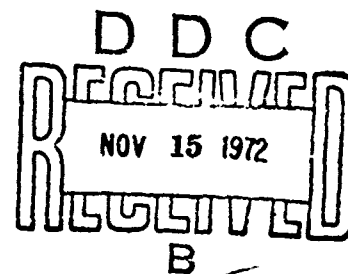


**ROFF**  
**A MANUSCRIPT PRINTING PROGRAM**  
**USER'S MANUAL**

Clifford E. Rhoades, Jr.

TECHNICAL REPORT NO. AFWL-TR-72-139

November 1972



**AIR FORCE WEAPONS LABORATORY**

**Air Force Systems Command**

**Kirtland Air Force Base**

**New Mexico**

Reproduced by  
**NATIONAL TECHNICAL**  
**INFORMATION SERVICE**  
U S Department of Commerce  
Springfield VA 22151

Approved for public release; distribution unlimited.

ACCESSION No.		
NTIS	White Section	<input checked="" type="checkbox"/>
DTIC	Both Section	<input type="checkbox"/>
WPA NUMBER		<input type="checkbox"/>
DISTRIBUTION		
BY		
DIS. & AVAIL. CODES		
DISC.	A. NTL. & WPA OF SPECIAL	
A		

AIR FORCE WEAPONS LABORATORY  
 Air Force Systems Command  
 Kirtland Air Force Base  
 New Mexico 87117

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Air Force Weapons Laboratory (DYS) Kirtland Air Force Base, New Mexico 87117		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE ROFF--A MANUSCRIPT PRINTING PROGRAM--USER'S MANUAL			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) December 1971 through March 1972			
5. AUTHOR(S) (First name, middle initial, last name) Clifford E. Rhoades, Jr.			
6. REPORT DATE November 1972		7a. TOTAL NO. OF PAGES 116	7b. NO. OF REFS ---
8a. CONTRACT OR GRANT NO.  b. PROJECT NO. 8809CF  c. Task 006003  d.		9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-72-139  9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY AFWL (DYS) Kirtland AFB, NM 87117	
13. ABSTRACT (Distribution Limitation Statement A)  This report is intended primarily as a user's manual for the ROFF manuscript printing system. ROFF is a computer program which produces esthetically pleasing manuscripts from punched card source texts. Both microfilm output and the generation of magnetic tape for off line printing on a Magnetic Tape/Selectric Typewriter (MT/ST) are supported. One of the important advantages of the ROFF system is the great ease with which revisions, additions and corrections can be made to draft memorandums and technical papers. This report is itself an example of a ROFF generated manuscript.			

DD FORM 1473  
1 NOV 65

I

UNCLASSIFIED  
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Manuscript printing Text editing Manuscript editing Information systems						

AFWL-TR-72-139

ROFF - A MANUSCRIPT PRINTING PROGRAM: USER'S MANUAL

Clifford E. Rhoades, Jr.

TECHNICAL REPORT NO. AFWL-TR-72-139

Approved for public release; distribution unlimited.

FOREWORD

This research was performed under Program Element 61101F, Project 8809CF, Task 006003.

Inclusive dates of research were December 1971 through March 1972. This report was submitted 1 August 1972 by Air Force Weapons Laboratory Project Officer Lt Clifford E. Rhoades, Jr., AFWL(DYS).

The author is grateful to Dr. Peter Crean and Dr. Stephen Fulling for making available earlier ROFF documentation. Their help has been invaluable in the preparation of this document. The author especially wishes to thank Dr. Fulling for correcting the original manuscript. Production of this report using ROFF was accomplished by the diligent persistence of Mr. Reuben Jamharian.

This technical report has been reviewed and is approved.

*Clifford E. Rhoades, Jr.*  
CLIFFORD E. RHOADES, JR.  
Lt, USAF  
Project Officer

*Thomas C. May*  
THOMAS C. MAY  
Major, USAF  
Chief, Simulation Branch

*Edward D. Young, Jr.*  
EDWARD D. YOUNG, JR.  
Lt Colonel, USAF  
Chief, Technology Division

ABSTRACT

(Distribution Limitation Statement A)

This report is intended primarily as a user's manual for the ROFF manuscript printing system. ROFF is a computer program which produces esthetically pleasing manuscripts from punched card source texts. Both microfilm output and the generation of magnetic tape for off line printing on a Magnetic Tape/ Selectric Typewriter (MT/ST) are supported. One of the important advantages of the ROFF system is the great ease with which revisions, additions and corrections can be made to draft memorandums and technical papers. This report is itself an example of a ROFF generated manuscript.



CONTENTS

<u>Section</u>	<u>page</u>
I     MANUSCRIPT PRINTING	1
II    EQUATION GENERATION	12
III   ROFF CHARACTER SET	17
IV    CONVERSION OF TAPE TO MT/ST CARTRIDGES	20
V     OFF LINE PRINTING ON A MT/ST MACHINE	21
APPENDIX	25

## SECTION I

## MANUSCRIPT PRINTING

Introduction

ROFF is a Fortran program for producing high quality printed documents with the computer. ROFF operates on an input deck of alphabetic text, produced on an ordinary 029 keypunch, and produces a printed copy in manuscript form. Since the 029 keypunches do not have any direct provision for entering lower case letters, all input to ROFF is upper case; ROFF changes upper case letters into lower case when appropriate. For example, all the letters in any sentence are converted to lower case, with the exception of the first one. The conversion may be overridden by means of special "escape" characters which control the mapping, but do not appear in the output. These are discussed below.

The output is formatted as the user wishes. He is able to start pages or paragraphs at will, produce blank lines, cause margins to be placed on the right side of the page, change the line length and indenting, and other functions of this sort. Footnotes may be indicated, entered, and automatically numbered; they are printed at the bottom of the current page.

These operations are all handled by "control words" which the user inserts in his output deck at the appropriate points.

Description

ROFF is a program which generates microfilm and 7-track binary magnetic tape for off line processing on an IBM Magnetic Tape / Selectric Typewriter (MT/ST) of arbitrary text in manuscript format. By the use of control words placed in the input text data set, the user may control the format of the document produced. Text data sets are card decks punched in EBCDIC as described below. Provisions are included for automatic lowercase letters, right margins on pages, page numbering (if desired) and numerous other formatting features.

### Character mapping

Input to ROFF is a card deck, generally of upper case letters and punctuation. The contents of the input deck are converted to lower case as follows:

1. The first letter on each sentence is left in upper case. All other letters are set to lower case. A sentence is defined to be a set of characters ending in an end-of-sentence mark (., !, ?, :, followed by an optional ), ], ", ', footnote indicator (0-8-2), or reference indicator (9-1)) and the end of a card or two blanks.

2. Upper case can be forced for the next alphabetic character encountered (A-Z only) by inserting a cent sign ( ¢ ) anywhere before the letter. Thus to capitalize a proper name in the middle of a sentence:

INPUT: IT IS A NICE DAY IN ¢PRINCETON

OUTPUT: It is a nice day in Princeton

The cent sign evaporates leaving no space in the output. The cent sign carries over all non-alphabetic characters.

3. Lower case may be forced at the beginning of a sentence by using a dollar sign ( \$ ).

4. An entire string of characters can be capitalized by preceding it with a circumflex ( ^ ). The effect of the circumflex is terminated by the next blank character in the input or by another circumflex.

5. Any string of characters can be underlined by preceding it with an underscore ( \_ ). The effect of the underscore is terminated by the next blank character in the input or by another underscore.

6. Arbitrary strikeouts may be created by using the at-sign ( @ ); the at is roughly equivalent to the backspace key on a regular typewriter. For example to make a not equal sign, use =@/ to produce ≠. If a sequence is to be overstruck, place all the at-signs together. No mapping of the overstriking characters occurs; control characters except for @ and — are printed in this instance.

7. Any percent sign ( % ) in the input is treated as a non-blank character, but vanishes on output. This is often useful as a place holder: if the space between two words is filled with percent signs, the program will not insert or delete any extra blanks between the words in the output. The percent signs can be used to reserve space for later insertion of special symbols.

8. Mistyped characters on the input cards may be 'erased' by using the — (numeric G) after the offending character. n —'s cause the last N characters to be replaced by the next N characters of input (control characters count towards N). Regular mapping occurs unless otherwise controlled.

9. An 0-8-2 punch (numeric T) indicates the position of a footnote reference number in the text; it will appear in the printout as [n], where n, the number of footnotes on the current page, is determined at execution time. The procedure for entering the footnote textual material is described in the section dealing with control words.

10. A 1-9 multipunch indicates the position of a deferred reference number in the text; it will appear in the printout as (n), where n is the number of references since the last set of references were printed. n is determined at execution time. The procedure for entering deferred reference text is described in the section dealing with control words. The deferred reference number is independent of the footnote reference number. (See 9. above)

11. A 2-9 multipunch followed by a digit n produces a footnote number which is n less than the most recent footnote number. e.g. If three footnotes have been indicated on the present page, a (2-9)1 will produce [2] in the output.

12. A 3-9 multipunch followed by a digit n produces a deferred reference number which is n less than the most recent deferred reference number. e.g. If twenty-five deferred references have been indicated, a (3-9)6 will produce (19) in the output, i.e. a citation to reference number 19, six less than the last reference number.

Special characters

The following printer characters are not available on the 029 keypunch but may be created by multipunching (hold down the MULT PCH key while striking several eettters). The MULT PCH key also gives numeric shift.

SYMBOL	MULTIPUNCHES
{	&0
}	-0
[	\$&0
]	(-0
~	-01
!	&-084
+	+ -0
^	81

Control words

The format of the output may be controlled by control cards. To distinguish control cards from the rest of the text, they have a unique format -- period in column 1, two letter abbreviation for the control word in columns 2 and 3, and sometimes an operand in columns 4-80. No other text may appear on the control card. Control words affect the printed format but are never printed themselves.

In this discussion, the word "break" associated with a control word will indicate that two input cards separated by the control card will not be run together, as they normally would in FILL mode. Thus at a break, all input text read so far will be printed out, and all following input text will appear on a new line of output.

"Default" means the value of the parameter that ROFF assumes if not otherwise specified.

- .PL n    page length  
           set output page length to n lines. Default and initial values are set to 48.
- .LL n    line length  
           break, set output line length to n characters. Default and initial values are set to 60.
- .SS      single space  
           break, enter single space mode. ROFF starts in single space mode.

- .DS      double space  
         break, print succeeding output double spaced.
- .NS      no spacing  
         break, do not space the carriage when  
         printing output lines.
- .CO      copy  
         enter mode in which all text (excepting  
         control words) is printed in upper case (no  
         mapping to lower case) and escape characters  
         have no effect.
- .MA      map  
         enter character mapping mode, the inverse of  
         copy. ROFF starts in map mode.
- .FI      fill  
         break, move words from the following cards as  
         necessary to place as many words as possible  
         on each line of output. ROFF starts in fill  
         mode.
- .NF      no fill  
         break, turn off fill mode. only mapping takes  
         place (if desired); no words are moved.
- .AD      adjust  
         break, turn on mode in which all text is  
         right justified by inserting blanks and  
         moving input words when necessary. ROFF  
         starts in adjust mode. When adjust is turned  
         on, so is fill.
- .NJ      no justification  
         break, turn off right justification of  
         margins. Nojust also turns off fill.
- .IN n    indent  
         break, print the following text with the left  
         margin indented n spaces from the normal  
         position. Default is n=0, which restores the  
         non-indenting.
- .UN n    undent  
         start the next line (only) n spaces to the  
         left of the current margin. Undent does not  
         change the current value of the indentation

nor will it move the print to the left of the natural margin.

- .PP n paragraph  
break, start a new paragraph with initial indentation n spaces relative to current indent value. If n is defaulted, use previous value for paragraph indenting. Initially n is 5. Capitalization is set on.
- .BR break  
break, set capital switch; the next input line is started on a new line.
- .SP n space  
break, insert n blank lines. Default: n=1. if the request cannot be satisfied on the current page, a skip to a new page executed first.
- .NE n need  
if n lines are left on the current page, no action is taken. Otherwise, break, and skip to a new page. Default: n=0.
- .PM n paging mode  
if n=1, print page numbers at the top of each page in arabic numerals. If n=2, print page numbers in lower case roman numerals. If n=0, don't print page numbers, but continue computing them. A change from roman to arabic and vice versa resets page number to 1. ROFF starts with n=1.
- .BP begin page  
break, start next line on new page. Capitalize first letter on new page.
- .PA n page  
break, start next line on a new page numbered n. Default: n=1. capitalize first letter on new page.
- .SK n skip  
at the first opportunity, skip n blank pages. Default: n=1. If further skips are encountered before previous ones are executed, the values of n are added, and all

executed at the first opportunity.

- .CE center  
break, center the input from the next card in the output line. The center switch turns itself off after the execution of one input card.
- .RA right adjust  
break, slide the text from the next input card over against the right margin. The right adjust switch turns itself off after the execution of one input card.
- .TRac translate  
henceforth, when the character a is encountered as the output is about to be printed, convert it to the character c. The characters are arbitrary and may be placed anywhere in the operand field. a may not be a blank. ROFF starts with '.TR % '.
- .RT revert  
return the transformation table set up by the .TR command to an identity transform with % going to blank.
- .CH /string1/string2/ change command  
change every occurrence of character string 'string1' to the character string 'string2'. String1 and string2 need not be the same length. Blanks within a string are significant[1]. If '/' appears within the character strings its role as a delimiter must be taken by any character not appearing in the strings. ROFF services twenty or less change commands simultaneously with the restriction that a string cannot exceed ten characters. If no operands are used, only previously entered changes are performed. The change command is rather time consuming when turned on.

[1] The character operands in the .CH and .TR commands are not mapped. To enter lower case letters an additional punch must be made on each letter: 0 for A-I, & for J-R, and - for S-Z.



- .NC      no change  
         turns off the change command. All changes entered are remembered and reinstituted by the next .CH command.
  
- .FN      begin footnote text  
         process the input cards to follow as the text for the m th footnote, where m is the number of footnote texts entered along with the current output page. The footnotes are stored on disk and printed at the lower portion of the page when the main body of text has printed. The first letter of the footnote text is capitalized unless otherwise controlled. The footnote is printed with no indentation in adjust mode and map mode unless controlled by control words entered within the footnote text itself. Output text is single spaced unless changed by the .FS command (see below).
  
- .FE      end footnote text  
         on the next card return to producing main text and return the program controls to their state before the .FN command[1].
  
- .FS      footnote spacing  
         set the footnote carriage control to provide the spacing currently in use, either single or double spacing. ROFF starts in single space mode.
  
- .CT      continuous footnote numbering  
         number the footnotes continuously from 1 throughout the text rather than resetting the number to 1 each page.
  
- .RF      begin deferred reference text  
         process the input cards to follow as text for

[1] The footnote indicator 0-8-2 and the footnote texts are numbered separately so many footnotes may be indicated before any footnote texts are entered. To enter several footnotes at one time, preface each by a .FN card and use only one .FE card after the last footnote text. If the footnotes will not fit on the current page, they carry over to the bottom of the next page.

the m th deferred reference, where m is the number of deferred references entered since the last .RP card (see below). The references are stored on disk and printed upon command (.RP). The first letter of the reference is capitalized unless otherwise controlled. The reference is printed with no indentation in adjust mode or map mode unless controlled by control words entered within the reference text itself. Output text is single spaced unless changed by the .RS command (see below).

- .RE end deferred reference text  
on the next card return to producing main text and return the program controls to their state before the .RF command.[1]
- .RP print deferred references  
space to new page, write heading REFERENCES and print all reference texts entered since the last call to .RP. Reset deferred reference number counter to 1.
- .RS reference spacing  
set the reference carriage control to provide the spacing currently in use, either single or double spacing. ROFF starts in single space mode.
- .EF end of file  
break, skip to the next page, terminate job. This should be the last card in the input deck.
- .HE heading  
causes the characters "xxx . . ." punched in columns 5-58 to appear (without mapping) at the top of each page (on the line with the page number, left justified) until cancelled

[1] The deferred reference indicator multipunch 1-9 and the reference texts are numbered separately and independently of footnote numbers, so that many references may be indicated before any reference texts are entered. To enter several references at one time, preface each by a .RF card and use only one .RE card after the last reference text.

by another .HE command (for which the character string may be blank).

.SF n set footnote counter  
set the footnote counter to n. (The first footnote will then be numbered n + 1.)

.FR change footnotes to references  
treat the footnotes as references. This allows the footnotes to be transferred from the bottom of the page to the end of the manuscript without repunching.

### Suggestions and Warnings

1. As a general rule, place each sentence on a separate card if running in fill mode. This makes editing the deck significantly easier.
2. A word cannot be run off the end of a card and onto the next input card. Also the @ and — features do not operate across a card boundary.
3. Only one overstrike is made for a given character.
4. The percent sign is very useful for controlling spaces when in fill or adjust mode. Its use can prevent the insertion of blanks and stop the elimination of blanks in the output line.
5. Only enter .RF and .FN text from the normal mode. Although footnotes may contain references and vice versa, the texts must be entered as .RF ... .RF ... .FN ... .FN, i.e. no overlapping of footnote and reference texts.

### General use of ROFF

To use ROFF, create the input deck as described in this manual, using control words and escape characters as needed. Remember to reset any parameters you desire that differ from the default values. The first page of output is not numbered and the second page is numbered as 1.

The load module for this program is stored on disk and available for general use. To run from this, submit the following deck:

JOBNAME,CM60000, ETC.

TASK CARD

COMMON,DYSROFF.

SWITCH,1. ONLY FOR MT/ST OUTPUT

REQUEST TAPE9,HI,,L. ETC. NEEDED FOR MT/ST ONLY

SWITCH,2. ONLY FOR MICROFILM OUTPUT

DYSROFF.

7/8/9

12/11/0/1/2/3/4/5/6/7/8/9 IN COLUMNS 1 AND 2

input deck

12/11/0/1/2/3/4/5/6/7/8/9 IN COLUMNS 1 AND 2

6/7/8/9

ROFF produces one single spaced page for every 50 cards of input. In two minutes 8000 cards of input can be processed to give 125 single spaced pages of output, of both microfilm and magnetic tape.

#### References

The major effort in devising the scheme for this program is due to Mr. J. Saltzer at MIT, who is responsible for specifying most of the basic commands used in ROFF. This particular version of ROFF is a modified form of an IBM OS/360 program written by Dr. Peter Crean as revised to incorporate equation writing by Dr. Stephen Fulling at Princeton.

## SECTION II

### EQUATION GENERATION

#### Introduction

EQROFF is a ROFF Fortran subprogram which formats equations and other material requiring alignment of several consecutive lines of print. From instructions punched sequentially on input cards it positions superscripts and subscripts, constructs fractions, and overlines expressions.

#### Mathematical equations

From instructions punched sequentially on the input cards EQROFF positions superscripts and subscripts in the output lines above and below the main line of the mathematical expression, constructs fractions, and draws lines over expressions.

1. Input cards for each line of an equation (or sequence of equations) must be preceded by the control card ".EQ" (equation). The last line must be followed by a control card (for example, ".PP" if a new paragraph is to begin). If no control statement is needed, use the dummy statement ".EE" (end of equation); the capital switch is then turned off. If the next line should begin with a capital letter, use .BR instead of .EE.

2. The following commands may make the testing of punched input less time-consuming in some circumstances:

.EO (equations only)      Process only equations, no text.

.NQ (no equations)      Process only text, no equations.

.AL (all input)      Cancel .EO or .NQ command.

3. Use of EQROFF inside a footnote is risky, as the subscripts or superscripts may appear on a different page from the rest of the equation.

4. The processing of equations is rather slow.

Control characters within an equation

1) EQROFF accepts and processes unchanged the following ROFF control characters:

a) ¢ This causes the next letter to be capitalized.

b) \$ This causes the next letter to be lower case.

c) ^ This causes the next string of letters to be capitalized.

d) \_ This causes the next string of characters to be underlined.

e) @ This causes the character following @ to overstrike the character preceding @. Any character (except - and @) may follow an @. This is useful for generating approximations to Greek letters, etc.[1] In particular, a ROFF or EQROFF control character may be used as the character following @; i.e., a@ to underline a single character a. In addition, if a single EQROFF control character is needed as part of an equation, & for example, it should be punched as %&.

f) % This forces a blank [see 3) below].

2) — (numeric G) causes the erasure of the previous character punched, thus permitting the correction of a misspelled character. Like ROFF, in EQROFF the — can erase control characters as well.

3) Blanks are always ignored. Thus the input card to EQROFF may have the various terms of an equation widely separated for ease of reading and editing. If a true blank is desired in the final output, a % sign should be used.

[1] To enter lower case letters following @ an additional punch must be made on each letter: 0 for A-I, & for J-R, and - for S-Z.

4) Normally an equation will consist of a centered or left justified expression followed by an optional right justified expression. This positioning is controlled as follows:

a) expression If an expression is not enclosed in control characters, it is written on the main equation line, starting two spaces in from the left margin.

b) (12-11)expression(12-11) This causes the expression within the (12-11)'s to be centered on the output line. This feature is useful for short expressions which look better when centered. Note that (12-11) is a multipunch (&-) on the keypunch.

c) `expression` This causes the expression within the ` 's (grave accents) to be right justified at the end of the manuscript line. This feature is useful for writing equation numbers.[1] Note that the ` is a multipunch (8-1) on the keypunch.

5) Superscripts, subscripts, and the numerators and denominators of fractions will appear on the lines above and below the main line. They are punched on the input cards at their natural locations in an expression as follows:

a) ?expression? This causes the expression enclosed within the ?'s to be written as a superscript (i.e., in the line above the main equation line.) Overlining is permitted within a superscript, but subscripts, superscripts, and fractions are not.

b) "expression" This causes the expression enclosed within the "'s to be written as a subscript (i.e., in the line below the main equation line). Overlining is permitted within a subscript, but subscripts, superscripts, and fractions are not.

[1] The deferred reference option in ROFF can be used to insert sequential equation numbers of the form (n) by punching `(9-1)%%` at the end of the EQROFF input card.

c) (0-8-2)superscript(0-8-2)subscript(0-8-2)  
This causes simultaneous super- and subscripting of expressions. The two expressions are left justified within the super/subscript expression. This feature is useful for limits of summations and integrals. Overlining is permitted within either term, but subscripts, superscripts, and fractions are not. Exception: The subscript control (") can be used within the superscript half of an (0-8-2) expression in order to write on the main and subscript lines simultaneously; similarly, ?'s may appear in the subscript term. Note that the multipunch (0-8-2) is the numeric T on the keypunch.

d) &numerator&denominator& This causes the first expression to be written as the numerator of a fraction, the second as the denominator. A bar is also written and the shorter term is centered within the fraction. Overlining is permitted within either term of a fraction, but subscripts, superscripts, and fractions are not.

6) #expression# This causes the expression enclosed within the #'s to be overlined. (Overlining is done by underlining the appropriate characters in the line above.) This feature is useful for writing square roots, e.g.,  $\sqrt{\text{term}}$ , and also for indicating complex conjugates, etc. Subscripts are permitted within the #'s, but superscripts and fractions are not.

Note that (12-11)'s and ^'s should not appear within the expressions described in 5) and 6).

#### Suggestions and warnings

- 1) Remember to leave space (by using %'s) for characters which must be added by hand, including superscripts inside fractions, etc.
- 2) If the last character of an expression to be underlined is also to be overstruck, type the overstrike (@x) before the ( ) which turns off the underlining.
- 3) If, in the output, the equation is scrambled or part of it is missing, check the entire input card



carefully to make sure that all the required control characters are present (e.g., three ampersands to every fraction). The relation of the error to the result may not be obvious. The program signals certain control character errors by placing a # in the line above the superscripts.

### An example

1) ROFF input cards:

```
.EQ
      f a ∞ % a a %    &    √ # X % + % Y #    % - % | Y |    &    √ # X % + % Y #    &    % E ? - α X ?
      % D X % = %      Ψ " P " ( X ? 2 ?    % - % Y ? 2 ? )    ~ ( I F % X % > @ _ % Y ) % % c % % `
.EQ
ba      "% L I - M " a X - - - @ > 0    a % F ( Y - X )    % = %    ϕ D " 1 " ( X )    b ` c % % `
.EE
```

Here the lower case letters stand for the following multipunches:

a for (0-8-2)    b for (12-11)    c for (9-1).

2) ROFF treatment of these cards:

last line of text above

$$\int_a^\infty \frac{\sqrt{x+y} - |y|}{\sqrt{x+y}} e^{-\alpha x} dx = \Psi \left( \frac{x^2}{p} - y^2 \right) \quad (\text{if } x \geq y) \quad (1)$$

$$\lim_{x \rightarrow 0} f(x) = D_1(x) \quad (2)$$

first line of text below

## SECTION III

## ROFF CHARACTER SET

GRAPHIC	CARD CODE	GRAPHIC	CARD CODE	GRAPHIC	CARD CODE
∇	12-9-8-7	∞	12-9-2	ψ	12-9-3
φ	12-9-4	TAB	12-9-5	¶	12-9-6
Λ	12-9-7	†	12-9-8-1	Π	12-9-8-2
π	12-9-8-3	‡	12-9-8-4	CR	12-9-8-5
ι	12-9-8-6	~	11-9-1	§	11-9-2
Ω	11-9-3	ø	11-9-4	BS	11-9-6
ℓ	11-9-7	Γ	11-9-8	Θ	11-9-8-1
J	11-9-8-2	ℒ	11-9-8-3	ℑ	11-9-8-4
‡	11-9-8-5	Σ	0-9-2	→	0-9-3
≡	0-9-4	α	0-9-5	Δ	0-9-6
≡	0-9-7	T	0-9-8	≈	0-9-8-1
α	12-0-9-1	β	12-0-9-2	ψ	12-0-9-3
φ	12-0-9-4	ε	12-0-9-5	ℓ	12-0-9-6
λ	12-0-9-7	η	12-0-9-8	ι	12-8-1
¢	12-8-2	.	12-8-3	<	12-8-4
(	12-8-5	+	12-8-6		12-8-7
ξ	12	j	12-11-9-1	κ	12-11-9-2
ω	12-11-9-3	μ	12-11-9-4	ν	12-11-9-5
ο	12-11-9-6	ρ	12-11-9-7	γ	12-11-9-8
θ	11-8-1	!	11-8-2	\$	11-8-3

AFWL-TR-72-139

*	11-8-4	)	11-8-5	;	11-8-6
—	11-8-7	-	11	/	0-1
σ	11-0-9-2	τ	11-0-9-3	ε	11-0-9-4
x	11-0-9-5	δ	11-0-9-6	χ	11-0-9-7
υ	11-0-9-8	ζ	0-8-1	,	0-8-3
%	0-8-4		0-8-5	>	0-8-6
?	0-8-7	f	12-11-0	✓	12-11-0-9-1
—	12-11-0-9-2	≡	12-11-0-9-3	\	12-11-0-9-4
±	12-11-0-9-5	∇	12-11-0-9-6	f	12-11-0-9-7
f	12-11-0-9-8	~	8-1	:	8-2
#	8-3	@	8-4	'	8-5
=	8-6	"	8-7	÷	12-0-8-1
a	12-0-1	b	12-0-2	c	12-0-3
d	12-0-4	e	12-0-5	f	12-0-6
g	12-0-7	h	12-0-8	i	12-0-9
.	12-11-8-1	j	12-11-1	k	12-11-2
l	12-11-3	m	12-11-4	n	12-11-5
o	12-11-6	p	12-11-7	q	12-11-8
r	12-11-9	~	11-0-1	s	11-0-2
t	11-0-3	u	11-0-4	v	11-0-5
w	11-0-6	x	11-0-7	y	11-0-8
z	11-0-9	0	12-11-0-8-1	1	12-11-0-1
2	12-11-0-2	3	12-11-0-3	4	12-11-0-4
5	12-11-0-5	6	12-11-0-6	7	12-11-0-7

AFWL-TR-72-139

8	12-11-0-8	9	12-11-0-9	'	12-11-0-8-2
[	12-11-0-8-3	\	12-11-0-8-4	]	12-11-0-8-5
^	12-11-0-8-6	+	12-11-0-8-7	{	12-0
A	12-1	B	12-2	C	12-3
D	12-4	E	12-5	F	12-6
G	12-7	H	12-8	I	12-9
}	11-0	J	11-1	K	11-2
L	11-3	M	11-4	N	11-5
O	11-6	P	11-7	Q	11-8
R	11-9	..	11-0-9-1	S	0-2
T	0-3	U	0-4	V	0-5
W	0-6	X	0-7	Y	0-8
Z	0-9	0	0	1	1
2	2	3	3	4	4
5	5	6	6	7	7
8	8	9	9		

## SECTION IV

### CONVERSION OF TAPE TO MT/ST CARTRIDGES

#### Initial

1. Obtain permission of operator of LITTON converter.
2. Turn the power switch to the "on" position. Switch is under cover.

#### Loading

1. Load file protected tape by hand making sure it is secure.
2. Reel tape to other holder making sure the tape follows the arrows.
3. Load cartridge by hand and make sure it is on firmly.
4. Press load button.
5. Select data file.
6. Press data file button.
7. Press transfer button.
8. If copying more than one file replace cartridge after it is rewound and then select next file.
9. Repeat steps 6, 7 and 8.

#### Unloading

1. To remove tape, hit rewind button.
2. Remove tape by hand
3. Remove cartridge.
4. Turn the power switch to the off position.

## SECTION V

### OFF LINE PRINTING ON A MT/ST MACHINE

#### Initial

1. Obtain permission of MT/ST secretary.
2. Remove light gray cover from machine if on.
3. Turn power switch on. Set spacing mode to single space.
4. Set paper guide at 0, right margin at 10, and left margin at 130.

#### Loading Cartridges

1. Set right control knob to L (or R) play and the left knob to auto.
2. Press unload to open plastic door.
3. Insert cartridge on L (or R) hub firmly.
4. Press load firmly.
5. Set reference number to 01.
6. Press search.
7. Press skip.
8. After loading paper, press start to run off the page.

#### Next Page

1. End of page is signaled by 5 carriage returns followed by a stop.
2. Change paper.

3. Advance reference number by one.
4. Press search.
5. Press start.

#### Unloading Cartridge

1. End of cartridge signaled immediately after a page change by a feed code followed by a stop. (Pressing start again will yield zzzzz followed by an error light).
2. To unload cartridge, press rewind.
3. Hold down unload and open the plastic door.
4. Gently remove cartridge from hub.

#### ERRORS

1. Error is signaled by lighting the error indicator light.
2. To recover, place right control knob in record L (or R).
3. Type the correct character.
4. Return right switch to play L (or R).
5. Press start to continue running off the page.

#### Greek Letters

1. Greek letters are signaled by a stop.
2. To continue, remove current selectric (roman) ball.
3. Replace with greek symbol ball.
4. Press start.
5. After stop, replace previous roman selectric ball.

Finish

1. Turn power off after removing cartridge.
2. Recover machine (if previously covered).
3. Record time and number of pages used (including spoiled) on log.



APPENDIX

LISTING OF ROFF CODE

```

OVLRLAY(SY,ROFF,J,J)
SUBROUTINE PR1 (JOYT,ISART,INLENG,IEO)
COMMON /CARDS/ NC,MES(5)
LOGICAL SECJY)
INTEGER GET
DIMENSION JOYT(80),IOATA(17)
DIMENSION ITRANS(4.96)
DATA NC/0/
DATA MES/3,M NUMBER OF CARDS READ BY ROFF /
DATA MES(5)/0/
DATA SECJY/.FALSE./
DATA ITRANS/4.96*0/
DATA ITRANS(2420)/0/
DATA ITRANS(2306)/1/
DATA ITRANS(2178)/2/
DATA ITRANS(2114)/3/
DATA ITRANS(2182)/4/
DATA ITRANS(2166)/5/
DATA ITRANS(2158)/6/
DATA ITRANS(2154)/7/
DATA ITRANS(2152)/8/
DATA ITRANS(2108)/9/
DATA ITRANS(2180)/10/
DATA ITRANS(2116)/11/
DATA ITRANS(2184)/12/
DATA ITRANS(2168)/13/
DATA ITRANS(2160)/14/
DATA ITRANS(2156)/15/
DATA ITRANS(3332)/16/
DATA ITRANS(1282)/17/
DATA ITRANS(1154)/18/
DATA ITRANS(1090)/19/
DATA ITRANS(1158)/20/
DATA ITRANS(1142)/21/
DATA ITRANS(1134)/22/
DATA ITRANS(1130)/23/
DATA ITRANS(1128)/24/
DATA ITRANS(1184)/25/
DATA ITRANS(1156)/26/
DATA ITRANS(1192)/27
DATA ITRANS(1160)/28/
DATA ITRANS(1144)/29/
DATA ITRANS(1136)/30/
DATA ITRANS(1132)/31/
DATA ITRANS(1196)/32/
DATA ITRANS(1100)/33/
DATA ITRANS(1142)/34/
DATA ITRANS(1108)/35/
DATA ITRANS(1146)/36/
DATA ITRANS(1130)/37/
DATA ITRANS(1222)/38/
DATA ITRANS(1118)/39/
DATA ITRANS(1116)/40/
DATA ITRANS(1172)/41/
DATA ITRANS(1144)/42/

```

ROFF	2
ROFF	3
ROFF	4
ROFF	5
ROFF	6
ROFF	7
ROFF	8
ROFF	9
ROFF	10
ROFF	11
ROFF	12
ROFF	13
ROFF	14
ROFF	15
ROFF	16
ROFF	17
ROFF	18
ROFF	19
ROFF	20
ROFF	21
ROFF	22
ROFF	23
ROFF	24
ROFF	25
ROFF	26
ROFF	27
ROFF	28
ROFF	29
ROFF	30
ROFF	31
ROFF	32
ROFF	33
ROFF	34
ROFF	35
ROFF	36
ROFF	37
ROFF	38
ROFF	39
ROFF	40
ROFF	41
ROFF	42
ROFF	43
ROFF	44
ROFF	45
ROFF	46
ROFF	47
ROFF	48
ROFF	49
ROFF	50
ROFF	51
ROFF	52
ROFF	53
ROFF	54
ROFF	55
ROFF	56

SUBROUTINE PRE

DATA ITRANS(590)/43/  
 DATA ITRANS(598)/44/  
 DATA ITRANS(532)/45/  
 DATA ITRANS(524)/46/  
 DATA ITRANS(520)/47/  
 DATA ITRANS(3844)/48/  
 DATA ITRANS(258)/49/  
 DATA ITRANS(150)/50/  
 DATA ITRANS(55)/51/  
 DATA ITRANS(34)/52/  
 DATA ITRANS(16)/53/  
 DATA ITRANS(15)/54/  
 DATA ITRANS(51)/55/  
 DATA ITRANS(4)/56/  
 DATA ITRANS(250)/57/  
 DATA ITRANS(152)/58/  
 DATA ITRANS(53)/59/  
 DATA ITRANS(35)/60/  
 DATA ITRANS(25)/61/  
 DATA ITRANS(12)/62/  
 DATA ITRANS(8)/63/  
 DATA ITRANS(1)/64/  
 DATA ITRANS(2518)/65/  
 DATA ITRANS(2590)/66/  
 DATA ITRANS(2526)/67/  
 DATA ITRANS(2594)/68/  
 DATA ITRANS(2578)/69/  
 DATA ITRANS(2570)/70/  
 DATA ITRANS(2566)/71/  
 DATA ITRANS(2564)/72/  
 DATA ITRANS(2307)/73/  
 DATA ITRANS(2179)/74/  
 DATA ITRANS(2115)/75/  
 DATA ITRANS(2583)/76/  
 DATA ITRANS(2567)/77/  
 DATA ITRANS(2559)/78/  
 DATA ITRANS(2555)/79/  
 DATA ITRANS(2549)/80/  
 DATA ITRANS(3530)/81/  
 DATA ITRANS(3202)/82/  
 DATA ITRANS(3138)/83/  
 DATA ITRANS(3106)/84/  
 DATA ITRANS(3590)/85/  
 DATA ITRANS(3582)/86/  
 DATA ITRANS(3578)/87/  
 DATA ITRANS(3576)/88/  
 DATA ITRANS(1283)/89/  
 DATA ITRANS(1155)/90/  
 DATA ITRANS(1591)/91/  
 DATA ITRANS(1559)/92/  
 DATA ITRANS(1543)/93/  
 DATA ITRANS(1535)/94/  
 DATA ITRANS(1531)/95/  
 DATA ITRANS(1525)/96/  
 DATA ITRANS(759)/97/

ROFF 57  
 ROFF 58  
 ROFF 59  
 ROFF 60  
 ROFF 61  
 ROFF 62  
 ROFF 63  
 ROFF 64  
 ROFF 65  
 ROFF 66  
 ROFF 67  
 ROFF 68  
 ROFF 69  
 ROFF 70  
 ROFF 71  
 ROFF 72  
 ROFF 73  
 ROFF 74  
 ROFF 75  
 ROFF 76  
 ROFF 77  
 ROFF 78  
 ROFF 79  
 ROFF 80  
 ROFF 81  
 ROFF 82  
 ROFF 83  
 ROFF 84  
 ROFF 85  
 ROFF 86  
 ROFF 87  
 ROFF 88  
 ROFF 89  
 ROFF 90  
 ROFF 91  
 ROFF 92  
 ROFF 93  
 ROFF 94  
 ROFF 95  
 ROFF 96  
 ROFF 97  
 ROFF 98  
 ROFF 99  
 ROFF 100  
 ROFF 101  
 ROFF 102  
 ROFF 103  
 ROFF 104  
 ROFF 105  
 ROFF 106  
 ROFF 107  
 ROFF 108  
 ROFF 109  
 ROFF 110  
 ROFF 111

Reproduced from  
 best available copy.

AFWL-TR-72-139

SUBROUTINE PRE

DATA ITRANS(1566)/98/	ROFF	112
DATA ITRANS(1502)/99/	ROFF	113
DATA ITRANS(1570)/100/	ROFF	114
DATA ITRANS(1554)/101/	ROFF	115
DATA ITRANS(1546)/102/	ROFF	116
DATA ITRANS(1542)/103/	ROFF	117
DATA ITRANS(1540)/104/	ROFF	118
DATA ITRANS(771)/105/	ROFF	119
DATA ITRANS(3.73)/106/	ROFF	120
DATA ITRANS(579)/107/	ROFF	121
DATA ITRANS(547)/108/	ROFF	122
DATA ITRANS(551)/109/	ROFF	123
DATA ITRANS(523)/110/	ROFF	124
DATA ITRANS(519)/111/	ROFF	125
DATA ITRANS(3585)/112/	ROFF	126
DATA ITRANS(3842)/113/	ROFF	127
DATA ITRANS(3714)/114/	ROFF	128
DATA ITRANS(3.50)/115/	ROFF	129
DATA ITRANS(3518)/116/	ROFF	130
DATA ITRANS(3502)/117/	ROFF	131
DATA ITRANS(3594)/118/	ROFF	132
DATA ITRANS(3590)/119/	ROFF	133
DATA ITRANS(3588)/120/	ROFF	134
DATA ITRANS(259)/121/	ROFF	135
DATA ITRANS(131)/122/	ROFF	136
DATA ITRANS(57)/123/	ROFF	137
DATA ITRANS(35)/124/	ROFF	138
DATA ITRANS(13)/125/	ROFF	139
DATA ITRANS(11)/126/	ROFF	140
DATA ITRANS(7)/127/	ROFF	141
DATA ITRANS(2519)/128/	ROFF	142
DATA ITRANS(2517)/129/	ROFF	143
DATA ITRANS(2589)/130/	ROFF	144
DATA ITRANS(2525)/131/	ROFF	145
DATA ITRANS(2593)/132/	ROFF	146
DATA ITRANS(2577)/133/	ROFF	147
DATA ITRANS(2569)/134/	ROFF	148
DATA ITRANS(2565)/135/	ROFF	149
DATA ITRANS(2563)/136/	ROFF	150
DATA ITRANS(2562)/137/	ROFF	151
DATA ITRANS(2591)/138/	ROFF	152
DATA ITRANS(2527)/139/	ROFF	153
DATA ITRANS(2595)/140/	ROFF	154
DATA ITRANS(2579)/141/	ROFF	155
DATA ITRANS(2571)/142/	ROFF	156
DATA ITRANS(2567)/143/	ROFF	157
DATA ITRANS(3331)/144/	ROFF	158
DATA ITRANS(3329)/145/	ROFF	159
DATA ITRANS(3261)/146/	ROFF	160
DATA ITRANS(3137)/147/	ROFF	161
DATA ITRANS(3105)/148/	ROFF	162
DATA ITRANS(3.89)/149/	ROFF	163
DATA ITRANS(3.81)/150/	ROFF	164
DATA ITRANS(3.77)/151/	ROFF	165
DATA ITRANS(3.75)/152/	ROFF	166

SUBROUTINE PRE

DATA ITRANS(3074)/153/	
DATA ITRANS(3203)/154/	ROFF 167
DATA ITRANS(3139)/155/	ROFF 168
DATA ITRANS(3107)/156/	ROFF 169
DATA ITRANS(3391)/157/	ROFF 170
DATA ITRANS(3383)/158/	ROFF 171
DATA ITRANS(3379)/159/	ROFF 172
DATA ITRANS(1795)/160/	ROFF 173
DATA ITRANS(1793)/161/	ROFF 174
DATA ITRANS(1565)/162/	ROFF 175
DATA ITRANS(1561)/163/	ROFF 176
DATA ITRANS(1569)/164/	ROFF 177
DATA ITRANS(1553)/165/	ROFF 178
DATA ITRANS(1545)/166/	ROFF 179
DATA ITRANS(1541)/167/	ROFF 180
DATA ITRANS(1539)/168/	ROFF 181
DATA ITRANS(1538)/169/	ROFF 182
DATA ITRANS(1567)/170/	ROFF 183
DATA ITRANS(1563)/171/	ROFF 184
DATA ITRANS(1571)/172/	ROFF 185
DATA ITRANS(1555)/173/	ROFF 186
DATA ITRANS(1547)/174/	ROFF 187
DATA ITRANS(1543)/175/	ROFF 188
DATA ITRANS(3843)/176/	ROFF 189
DATA ITRANS(3841)/177/	ROFF 190
DATA ITRANS(313)/178/	ROFF 191
DATA ITRANS(3549)/179/	ROFF 192
DATA ITRANS(3517)/180/	ROFF 193
DATA ITRANS(3561)/181/	ROFF 194
DATA ITRANS(3593)/182/	ROFF 195
DATA ITRANS(3589)/183/	ROFF 196
DATA ITRANS(3587)/184/	ROFF 197
DATA ITRANS(3586)/185/	ROFF 198
DATA ITRANS(3715)/186/	ROFF 199
DATA ITRANS(3551)/187/	ROFF 200
DATA ITRANS(3519)/188/	ROFF 201
DATA ITRANS(3563)/189/	ROFF 202
DATA ITRANS(3595)/190/	ROFF 203
DATA ITRANS(3591)/191/	ROFF 204
DATA ITRANS(2561)/192/	ROFF 205
DATA ITRANS(2365)/193/	ROFF 206
DATA ITRANS(2177)/194/	ROFF 207
DATA ITRANS(2113)/195/	ROFF 208
DATA ITRANS(2081)/196/	ROFF 209
DATA ITRANS(2165)/197/	ROFF 210
DATA ITRANS(2057)/198/	ROFF 211
DATA ITRANS(2153)/199/	ROFF 212
DATA ITRANS(2151)/200/	ROFF 213
DATA ITRANS(2150)/201/	ROFF 214
DATA ITRANS(2592)/202/	ROFF 215
DATA ITRANS(2528)/203/	ROFF 216
DATA ITRANS(2596)/204/	ROFF 217
DATA ITRANS(2583)/205/	ROFF 218
DATA ITRANS(2572)/206/	ROFF 219
DATA ITRANS(2568)/207/	ROFF 220
	ROFF 221

AFWL-TR-72-139

SUBROUTINE PRE

DATA ITRANS(1,37)/208/  
 DATA ITRANS(1281)/209/  
 DATA ITRANS(1153)/210/  
 DATA ITRANS(1,89)/211/  
 DATA ITRANS(1,57)/212/  
 DATA ITRANS(1,41)/213/  
 DATA ITRANS(1,33)/214/  
 DATA ITRANS(1,29)/215/  
 DATA ITRANS(1,27)/216/  
 DATA ITRANS(1,26)/217/  
 DATA ITRANS(3264)/218/  
 DATA ITRANS(3140)/219/  
 DATA ITRANS(3108)/220/  
 DATA ITRANS(3,92)/221/  
 DATA ITRANS(3,84)/222/  
 DATA ITRANS(3,80)/223/  
 DATA ITRANS(643)/224/  
 DATA ITRANS(1794)/225/  
 DATA ITRANS(5,1)/226/  
 DATA ITRANS(577)/227/  
 DATA ITRANS(5,5)/228/  
 DATA ITRANS(5,29)/229/  
 DATA ITRANS(5,21)/230/  
 DATA ITRANS(5,7)/231/  
 DATA ITRANS(5,15)/232/  
 DATA ITRANS(5,14)/233/  
 DATA ITRANS(1068)/234/  
 DATA ITRANS(1,04)/235/  
 DATA ITRANS(1,72)/236/  
 DATA ITRANS(1,56)/237/  
 DATA ITRANS(1,48)/238/  
 DATA ITRANS(1,44)/239/  
 DATA ITRANS(5,13)/240/  
 DATA ITRANS(2,7)/241/  
 DATA ITRANS(1,29)/242/  
 DATA ITRANS(5,)/243/  
 DATA ITRANS(33)/244/  
 DATA ITRANS(17)/245/  
 DATA ITRANS(9)/246/  
 DATA ITRANS(5)/247/  
 DATA ITRANS(3)/248/  
 DATA ITRANS(2)/249/  
 DATA ITRANS(3,10)/250/  
 DATA ITRANS(3,52)/251/  
 DATA ITRANS(3,20)/252/  
 DATA ITRANS(3,504)/253/  
 DATA ITRANS(3,96)/254/  
 DATA ITRANS(3,92)/255/

IF (SECOND) GO TO 5  
 SECOND=.TRUE.  
 IFET=GET(\$INPUT)  
 CONTINUE  
 CALL PIN(IDATA,16,IFET,IEO)  
 IF (IEO.EQ.1) RETURN

POFF 222  
 ROFF 223  
 ROFF 224  
 ROFF 225  
 ROFF 226  
 ROFF 227  
 ROFF 228  
 ROFF 229  
 ROFF 230  
 ROFF 231  
 ROFF 232  
 ROFF 233  
 ROFF 234  
 ROFF 235  
 ROFF 236  
 ROFF 237  
 ROFF 238  
 ROFF 239  
 ROFF 240  
 ROFF 241  
 ROFF 242  
 ROFF 243  
 ROFF 244  
 ROFF 245  
 ROFF 246  
 ROFF 247  
 ROFF 248  
 ROFF 249  
 ROFF 250  
 ROFF 251  
 ROFF 252  
 ROFF 253  
 ROFF 254  
 ROFF 255  
 ROFF 256  
 ROFF 257  
 ROFF 258  
 ROFF 259  
 ROFF 260  
 ROFF 261  
 ROFF 262  
 ROFF 263  
 ROFF 264  
 ROFF 265  
 ROFF 266  
 ROFF 267  
 ROFF 268  
 ROFF 269  
 ROFF 270  
 ROFF 271  
 ROFF 272  
 ROFF 273  
 ROFF 274  
 ROFF 275  
 ROFF 276

AFVL-TR-72-139

SUBROUTINE PRE

	IS=ISART-1	ROFF	277
	NC=NC+1	ROFF	278
	DO 8 N=1,10	ROFF	279
	IWD=IOWATA(N)	ROFF	280
	DO 7 K=1,5	ROFF	281
	IN=SHIFT(IWD,12*K).AND.7777B	ROFF	282
	IS=IS+1	ROFF	283
	JOUT(IS)=IIRANS(IN+1)	ROFF	284
	IF (IS.EQ.IN.ENG) RETURN	ROFF	285
7	CONTINUE	ROFF	286
3	CONTINUE	ROFF	287
	END	ROFF	288

AFWL-TR-72-139

PROGRAM	IDENT	GETBAS	ROFF	289
LENGT4				
BLOCKS				
PROGRAM*	LOCAL			
ENTRY POINTS				
000000 GET				
EXTERNAL SYMBOLS				
GETBA				
	ENTRY	GET	ROFF	290
	EXT	GETBA	ROFF	291
GET	BSS	1	ROFF	292
*	FUNCTION GET(7LFILENAME)		ROFF	293
*	GET * FET ADDRESS		ROFF	294
	SA1	A1	ROFF	295
	SB2	X1	ROFF	296
	SB2	80-82	ROFF	297
	RJ	GETBA	ROFF	298
	SX6	82	ROFF	299
	LT	80,82,GET1	ROFF	300
	SX6	80	ROFF	301
SET1	EQ	SET	ROFF	302
	END		ROFF	303

UNUSED STORAGE                      15 STATEMENTS                      3 SYMBOLS



IDENT	PINS	ROFF	304
PROGRAM LENGTH			
BLOCKS			
PROGRAM LOCAL			
ENTRY POINTS			
000000 PIN			
ENTRY PIN		ROFF	305
* THIS ROUTINE READS N WORDS FROM INPUT FILE BUFFER AND STARTS		ROFF	306
* I/O IF EMPTY RETURNS FLAG IF EOR		ROFF	307
PIN BSS 1		ROFF	308
* SUBROUTINE PIN(DATA,NUMBER ,FETADD,IEO)		ROFF	309
* IEO=1 END OF LOGICAL RECORD		ROFF	310
* IEO=2 NORMAL END		ROFF	311
SB7 1		ROFF	312
SA1 A1		ROFF	313
SB1 X1	PICK UP DATA WORD ADDRESS	ROFF	314
SA1 A1+B7		ROFF	315
SA5 X1	LOAD NUMBER OF WORDS DESIRED	ROFF	316
SB2 X5	B2 IS THE NUMBER	ROFF	317
SX0 B7	X0=1	ROFF	318
SA1 A1+B7	FET ADDRESS	ROFF	319
SA2 A1+B7		ROFF	320
SB4 X2	IEO ADDRESS	ROFF	321
SA1 X1	X1 PICK IT UP	ROFF	322
SB6 X1	SAVE FET ADDRESS FOR READ	ROFF	323
SA3 X1+3	READ OUT	ROFF	324
IX7 X5-X0	THIS ELIMINATES ZERO CHECK LATER	ROFF	325
SA4 A3-B7	READ IN	ROFF	326
SA5 A3+B7	READ LIMIT	ROFF	327
SX5 X5	MASK OUT ALL BUT LIMIT SET X5=LIMIT	ROFF	328
PIN1 BSS 0		ROFF	329
IX1 X5-X3		ROFF	330
VZ X1,PIN2	SENSE OUT NOT LIMIT	ROFF	331
SA1 A4-B7	READ FIRST	ROFF	332
SX3 X1	OUT=FIRST	ROFF	333
PIN2 IX1	X4-X3	ROFF	334
ZR X1,READ	LOOK FOR OUT=IN	ROFF	335
PIN3 IX7	X7-X0	ROFF	336
SA2 X3	DECREMENT DATA COUNT	ROFF	337
SX3 X3+B7	READ DATA WORD AT OUT	ROFF	338
	INCREMENT OUT	ROFF	339
	OUT MUST BE SET TO ONE MORE THAN THE ADDRESS OF	ROFF	340
	OF WORD LAST TRANSMITTED	ROFF	341
3X6 X2		ROFF	342
SA6 B1	STORE AS REQUESTED	ROFF	343
NG X7,PINEND	SENS END OF TRANSFER	ROFF	344
SB1 B1+B7	INCREMENT TEMP LOCATION	ROFF	345
EQ PIV1	LOOP UNTIL NO MORE DATA	ROFF	346
PINEND BX5	STORE UPDATE OUT	ROFF	347
SA6 A3	DONE	ROFF	348
SX7 2		ROFF	349
SA7 B4	STORE 2 IN IEO	ROFF	350
EQ PIN	EXIT	ROFF	350

* THIS ROUTINE CALLS CIO WITH RECALL FOR BINARY READ			ROFF	351
READ	8X6	X3	ROFF	352
	SX1	86 GET BACK FET ADDRESS	ROFF	353
	SX3	3RCIO*108+2	ROFF	354
	SA6	A3 STORE OUT	ROFF	355
	LX3	39 4LCIOP	ROFF	356
	SA2	X1 FET FWA	ROFF	357
*	CHECK IF PREVIOUS OPERATION LEAD TO EOR		ROFF	358
	3X1	X3+X1 24/4LCIOP,36/FET	ROFF	359
	4X3	42	ROFF	360
	3X6	-X3*X2 GET STATUS	ROFF	361
	AX6	4 RIGHT SHIFT	ROFF	362
	ZR	X6,NO NO END OF RECORD	ROFF	363
	EQ	EQF SORRY	ROFF	364
READB	EQU	12J	ROFF	365
NO	BSS	;	ROFF	366
	3X6	X3*X2 MASK OUT STATUS	ROFF	367
	SX3	READB BINARY READ	ROFF	368
	3X6	X6+X3 42/LFN,18/READB	ROFF	369
	SA6	A2 STORE IN FET	ROFF	370
	3X6	X1 PREPARE CALL	ROFF	371
+	SA1	B7 CALL	ROFF	372
	VZ	X1,*	ROFF	373
	SA6	B7 CALL CIO	ROFF	374
+	SA1	B7	ROFF	375
	VZ	X1,* WAIT FOR MTR TO AWAKE	ROFF	376
	SA4	A4 GET NEW IN	ROFF	377
	SA3	A3 GET NE M OUT	ROFF	378
	EQ	PIV3 CHECK FOR MORE	ROFF	379
EOF	SX7	X0	ROFF	380
	SA7	B4	ROFF	381
	EQ	PIV	ROFF	382
	END		ROFF	383

UNUSED STORAGE                      80 STATEMENTS                      9 SYMBOLS

## PROGRAM ROFF

PROGRAM ROFF(INPUT=03108,TAPE9=0,FILMPL=03108,TAPE3=0,TAPE4=0)	ROFF	384
A F W L ( S Y S )	ROFF	385
VERSION OF 6 APRIL 1972	ROFF	386
IMPLICIT INTEGER (A-Z)	ROFF	387
INTEGER ADOFT,ADREF,ATCTR,ATSIGN,BADCTR,BANK,BN1,BUFFL,CC,	ROFF	388
1CCC,CCHOLD,CCSV,GEN,CENT,CFLX,COLON,COLUMN,CSAVE,D,DOLLAR,DUH,EQU	ROFF	389
2,EQU,EXCLAM,FDC,FLIN,FPCG,FTLINZ,FTOVER,FTREC,HEAD,HZERO,OLENG,OUT	ROFF	390
3,OVLIN,OB2,PAGEL,PAGEN,PAGENO,PAGES,PCG,PCGSV,PERGEN,PERIOD,PLUS,	ROFF	391
4PH,POS,PP,PPT,MP,QM,QUOTE1,QUOTE2,RBRACE,RBRACK,RCC,REFREC,RPAREN,	ROFF	392
5RPGC,SAVE,SAVE0,STATE,U,ULINE,USCORE,X,Z,ZERO,Z4	ROFF	393
COMMON /INBUF/ IN(99),ULINE(99),PRJ,INLENG,INL1	ROFF	394
COMMON /OUTBUF/ OUT(130),OVLIN(130),BUFFL,OVERSH,NWORD,OLENG,PSH,	ROFF	395
1LENMAX	ROFF	396
COMMON /EQBUF/ EQU(200,4),LMIN,LMAX,EQSW	ROFF	397
COMMON /OPARM/ CC,PCG,INDENT,PAGENO,LINECT,PAGEL,PHONSW,RNUMSW	ROFF	398
COMMON /FEET/ U,NREC,NFOOT,FTREC,FTNOTE,NFOOTP,FTOVER,FTLINZ,CTFN	ROFF	399
COMMON /SWITCH/ ADJW,FILLSW	ROFF	400
COMMON /SR/ COLUMN,INL2	ROFF	401
COMMON /SR3/ ITAB(256)	ROFF	402
COMMON /SR4/ SPELW	ROFF	403
COMMON /SPW/ SP	ROFF	404
COMMON /CARDS/NC	ROFF	405
LOGICAL EQSW,OVERSH,PSH,PRU,PHONSW,RNUMSW,CTFN,FTNOTE	ROFF	406
LOGICAL FTINS,FLAGSV(9),PRMORE,ASV,FSV	ROFF	407
DIMENSION SAVL(264),SAVED(264),CSAVE(4),INFAKE(130),INHOLD(81)	ROFF	408
EQUIVALENCE (SAVE(1),OUT(1)),(IN(1),INFAKE(1))	ROFF	409
COMMON /FLINK/ FLIN(131),HEAD(54),IDJM(6)	ROFF	410
COMMON /SKIPL/ PAGES	ROFF	411
CENTER INPUT LINE STARTS OFF	ROFF	412
	ROFF	413
	ROFF	414
	ROFF	415
SETTINGS FOR PARAMETERS	ROFF	416
SWITCHES FOR DEFAULTS, ETC.	ROFF	417
	ROFF	418
	ROFF	419
SET UP THE INITIAL VALUES OF FOOTNOTE AND REFERENCE CARRIAGE CONTR	ROFF	420
INPUT LINE LENGTH	ROFF	421
CARRIAGE CONTROL INDICATOR	ROFF	422
1 = SINGLE SPACE, 2 = DOUBLE SPACE	ROFF	423
DEFAULT LINE SPACING IS SINGLE	ROFF	424
INITIAL PARAGRAPH INDENT VALUE	ROFF	425
INITIALIZE REFERENCE COUNTER	ROFF	426
PAGING MODE ON TO START	ROFF	427
RIGHT-ADJUST SW ON TO START	ROFF	428
SWITCH FOR FILL MODE	ROFF	429
UNDERSCORE OFF TO START	ROFF	430
PRINT UNDERSCORE LINE OFF TO START	ROFF	431
OUTPUT OVERSTRIKE LINE OFF TO START	ROFF	432
CAPITALIZE FIRST WORD	ROFF	433
SWITCH FOR CAPITALIZING A WHOLE WORD	ROFF	434
COPY DIRECTLY, DEFAULT IS NO	ROFF	435
LOGICAL RIGHT,CENTER,SPELW,REFING,ADJW,FILLSW,USW,COPYSW,HOLDW,N	ROFF	436
1QSW,CCSW,EUSW,CAPSW,ALLCAP,ADSA,FISA,PSA,FRSW,SP	ROFF	437
DIMENSION IDJT(24)	ROFF	438

# AFWL-TR-72-139

PROGRAM ROFF

DATA HZERO/0/	ROFF	439
DATA DUM/0/,REFREQ/0/,FCG/1/,RCC/1/,PP/5/,NREF/0/,NREFP/0/,-Z/0/	ROFF	440
DATA ATCTR/0/.BAGCTR/0/	ROFF	441
DATA RIGHT,CENTER,REFING,USM,CAPSM,ALLCAP/.F.,.F.,.F.,.F.,.T.,.F./	ROFF	442
DATA COPYSM,HULDSM,NQSM,DCSA,EOSM,PSA,FRSM/.F.,.F.,.F.,.F.,.F.,.F./	ROFF	443
1.,.F./	ROFF	444

## TABLE OF NUMERICAL VALUES FOR GRAPHICS

DATA LREF/49/	ROFF	445
DATA LPAREN/77/	ROFF	446
DATA RBRACK/189/	ROFF	447
DATA LBRACK/137/	ROFF	448
DATA O82/224/	ROFF	449
DATA LNOT/95/	ROFF	450
DATA LETT/233/	ROFF	451
DATA LETTRY/232/	ROFF	452
DATA LETTRX/231/	ROFF	453
DATA LETTRM/230/	ROFF	454
DATA LETTRV/229/	ROFF	455
DATA LETTRU/228/	ROFF	456
DATA LETTRT/227/	ROFF	457
DATA LETTRS/226/	ROFF	458
DATA LETTRR/217/	ROFF	459
DATA LETTRQ/216/	ROFF	460
DATA LETTRP/215/	ROFF	461
DATA LETTRO/214/	ROFF	462
DATA LETTRN/213/	ROFF	463
DATA LETTRM/212/	ROFF	464
DATA LETTRL/211/	ROFF	465
DATA LETTRK/210/	ROFF	466
DATA LETTRJ/209/	ROFF	467
DATA LETTRI/208/	ROFF	468
DATA LETTRH/207/	ROFF	469
DATA LETTRG/206/	ROFF	470
DATA LETTRF/205/	ROFF	471
DATA LETTRE/204/	ROFF	472
DATA LETTRD/203/	ROFF	473
DATA LETTRC/202/	ROFF	474
DATA LETTRB/201/	ROFF	475
DATA LETTRA/200/	ROFF	476
DATA USCORE/199/	ROFF	477
DATA QUOTE2/127/	ROFF	478
DATA QUOTE1/125/	ROFF	479
DATA QM/111/	ROFF	480
DATA PERIOU/77/	ROFF	481
DATA PERCEN/1.8/	ROFF	482
DATA RPAREN/93/	ROFF	483
DATA XCLAM/3./	ROFF	484
DATA DOLLAR/31/	ROFF	485
DATA COLON/12./	ROFF	486
DATA FLEX/19./	ROFF	487
DATA CENT/74/	ROFF	488
DATA BLANK/6./	ROFF	489
DATA ATSIGN/124/	ROFF	490
	ROFF	491
	ROFF	492
	ROFF	493

# AFWL-TR-72-139

PROGRAM ROFF

DATA RBRACE/2.8/,LBRACE/192/	ROFF	494
DATA ADJFT/53/,ADDREF/51/	ROFF	495
DATA LBLANK,PLUS,ZERO/1H,14+,1H0/	ROFF	496
DATA FTING/,FALSE./	ROFF	497
INITIALIZE THE SPELLING CHANGE COUNTER	ROFF	498
SPELSH=.FALSE.	ROFF	499
SP=.FALSE.	ROFF	500
FPGC=LBLANK	ROFF	501
RPCG=LBLANK	ROFF	502
PCC=LBLANK	ROFF	503
COLUMN=0	ROFF	504
ADSW=.TRUE.	ROFF	505
FILLSW=.TRUE.	ROFF	506
	ROFF	507
	ROFF	508
INITIALIZE TRANSLATE TABLE FOR FINAL OUTPUT STAGE	ROFF	509
TO START WITH, ALL CHARACTERS GO TO THEMSELVES	ROFF	510
EXCEPT FOR O/L SIGN, WHICH GOES BLANK	ROFF	511
	ROFF	512
	ROFF	513
CALL INITTR (INFAK=)	ROFF	514
	ROFF	515
HERE WE GO	ROFF	516
	ROFF	517
SET CHARACTERS	ROFF	518
	ROFF	519
DO 1 I=1,256	ROFF	520
ITAB(I)=I	ROFF	521
	ROFF	522
LETTERS	ROFF	523
DO 2 I=193,231	ROFF	524
ITAB(I)=I-64	ROFF	525
DO 3 I=204,217	ROFF	526
ITAB(I)=I-64	ROFF	527
DO 4 I=226,233	ROFF	528
ITAB(I)=I-64	ROFF	529
DO 5 I=1,54	ROFF	530
HEAD(I)=BLANK	ROFF	531
	ROFF	532
	ROFF	533
THIS IS THE START OF THE MAIN LOOP.	ROFF	534
THE INPUT LINE IS READ HERE	ROFF	535
	ROFF	536
CONTINUE	ROFF	537
CALL PRE (IN,1,INLENG,IEO)	ROFF	538
GO TO (130,7), IEO	ROFF	539
CONTINUE	ROFF	540
INLENG=INLENG+1	ROFF	541
CHECK FOR CONTROL WORD	ROFF	542
IF (I(1).EQ.PERIOD) GO TO 55	ROFF	543
IF (EOSW.AND..NOT.EQSW) GO TO 6	ROFF	544
IF (NQS.W.AND..EQSW) GO TO 6	ROFF	545
IF IN DIRECT COPY MODE, SKIP CONVERSION, GO TO OUTPUT	ROFF	546
J=80	ROFF	547
IF (COPYSW) GO TO 26	ROFF	548

Reproduced from  
best available copy.

PROGRAM ROFF

1		ROFF	549
2		ROFF	550
3	CONVERT THE LINE	ROFF	551
4		ROFF	552
5	HANDLE CAPITALIZATION AND OTHER SPECIAL PROBLEMS	ROFF	553
6	IF (.NOT.EQSW) CALL CRRECT	ROFF	554
7	I=1	ROFF	555
8	J=1	ROFF	556
9	IF (I.GE.INLI) GO TO 28	ROFF	557
10	ITEMP=IN(I)	ROFF	558
11	IF (ATCTR.GT.J) GO TO 13	ROFF	559
12	JTEMP=IN(I+1)	ROFF	560
13	IF (ITEMP.LT.129.OR.ITEMP.GE.240) GO TO 10	ROFF	561
14	SEE IF SPECIAL CHARS OR NUMBERS	ROFF	562
15	IF (ITEMP.GT.169.AND.ITEMP.LT.193) GO TO 10	ROFF	563
16	NO, SO CONVERT IF NECESSARY	ROFF	564
17	IF (ITEMP.EQ.182) GO TO 18	ROFF	565
18	IN(J)=IN(I)	ROFF	566
19	IF (.NOT.CAPSW.AND..NOT.ALLCAP) IN(J)=ITAB(ITEMP)	ROFF	567
20	ARE WE (NOT) ING	ROFF	568
21	IF (USH) ULINE(J)=USCORE	ROFF	569
22	CAPSW=.FALSE.	ROFF	570
23	I=I+1	ROFF	571
24	J=J+1	ROFF	572
25	GO TO 9	ROFF	573
26		ROFF	574
27	SPECIAL CHARACTERS COME HERE.	ROFF	575
28		ROFF	576
29	CONTINUE	ROFF	577
30		ROFF	578
31		ROFF	579
32	IF (ITEMP.EQ.LREF) GO TO 23	ROFF	580
33	IF (ITEMP.EQ.ADDFT) GO TO 24	ROFF	581
34	IF (ITEMP.EQ.ADDREF) GO TO 25	ROFF	582
35	IF (ITEMP.EQ.BLANK) GO TO 11	ROFF	583
36		ROFF	584
37	TURN OF UNDERSCORE SW IF BLANK	ROFF	585
38	USH=.FALSE.	ROFF	586
39	TURN OFF ALLCAP	ROFF	587
40	ALLCAP=.FALSE.	ROFF	588
41	GO TO 12	ROFF	589
42	CONTINUE	ROFF	590
43	NOT MEANS BACKSPACE FOR ERASURE	ROFF	591
44	END SENTENCE PUNCTUATION	ROFF	592
45	IF (ITEMP.EQ.QM.OR.ITEMP.EQ.EXCLAM.OR.ITEMP.EQ.PERIOD.OR.ITEMP.EQ.	ROFF	593
46	LCOLON) GO TO 14	ROFF	594
47	AT SIGN MEANS BACKSPACE	ROFF	595
48	IF (ITEMP.EQ.ATSIGN) GO TO 22	ROFF	596
49	CENTS MEANS CAPITALIZE NEXT CHAR, AND BLANK SELF	ROFF	597
50	IF (ITEMP.EQ.CENT) GO TO 17	ROFF	598
51	IS IT TO UNDERSCORE	ROFF	599
52	IF (ITEMP.EQ.JSCORE) GO TO 19	ROFF	600
53	CFX MEANS CAPITALIZE NEXT WORD AND BLANK SELF	ROFF	601
54	IF (ITEMP.EQ.FLEX) GO TO 21	ROFF	602
55	IS IT \$ TO FORCE LOWER CASE	ROFF	603

AFWL-TR-72-139

	PROGRAM	ROFF	
	IF (ITEMP.EQ.DOLLAR) GO TO 20	ROFF	604
3	DO WE NEED AN UNDERSCORE	ROFF	605
	IF (USW) ULINE=(J)=USCORE	ROFF	606
	IF (ITEMP.GT.256.OR.ITEMP.LE.0) ITEM=124	ROFF	607
3	ANYTHING ELSE IS MAPPED	ROFF	608
12	IN(J)=ITAB(IT=MP)	ROFF	609
	J=J+1	ROFF	610
	I=I+1	ROFF	611
	GO TO 9	ROFF	612
3		ROFF	613
3	INSERT AT SIGN HANDLER HERE	ROFF	614
3		ROFF	615
13	JAT=J-ATCTR	ROFF	616
	ULINE(JAT)=IT=MP	ROFF	617
	ATCTR=ATCTR-1	ROFF	618
	I=I+1	ROFF	619
	GO TO 9	ROFF	620
3		ROFF	621
3	*, HERE. NOCONV, SET CAPSW, KILL UNDERSCORE	ROFF	622
14	I=I+1	ROFF	623
	IN(J)=ITEMP	ROFF	624
	IF (USW) ULINE=(J)=USCORE	ROFF	625
	J=J+1	ROFF	626
3		ROFF	627
3	IS THIS THE END OF THE SENTENCE	ROFF	628
	IF (JTEMP.EQ.BLANK) GO TO 15	ROFF	629
	IF (JTEMP.EQ.082.OR.JTEMP.EQ.LREF.OR.JTEMP.EQ.ADDFT.OR.JTEMP.EQ.ADDREF) GO TO 15	ROFF	630
	IF (JTEMP.NE.PAREN.AND.JTEMP.NE.QUOTE1.AND.JTEMP.NE.QUOTE2.AND.JTEMP.NE.RBRACE) GO TO 9	ROFF	631
	IN(J)=IN(I)	ROFF	632
	I=I+1	ROFF	633
	J=J+1	ROFF	634
15	CONTINUE	ROFF	635
	IF (IN(I).NE.BLANK.OR.IN(I+1).NE.BLANK) GO TO 9	ROFF	636
	IN(J)=1	ROFF	637
	IN(J+1)=BLANK	ROFF	638
	I=I+2	ROFF	639
	J=J+2	ROFF	640
	INL1=MAXD(INL1,J)	ROFF	641
16	CAPSW=.TRUE.	ROFF	642
	USW=.FALSE.	ROFF	643
	ALLCAP=.FALSE.	ROFF	644
	IF ((IN(I).EQ.082.OR.IN(I).EQ.LREF).AND.IN(I+1).EQ.BLANK) SP=.TRUE	ROFF	645
	1. IF ((IN(I).EQ.ADDFT.OR.IN(I).EQ.ADDREF).AND.IN(I+2).EQ.BLANK) SP=.TRUE.	ROFF	646
	GO TO 9	ROFF	647
3	SENTS HERE	ROFF	648
17	CONTINUE	ROFF	649
	CAPSW=.TRUE.	ROFF	650
	ALLCAP=.FALSE.	ROFF	651
	I=I+1	ROFF	652
	GO TO 9	ROFF	653
3	FOOTNOTE STUFF HERE	ROFF	654
		ROFF	655
		ROFF	656
		ROFF	657
		ROFF	658

Reproduced from  
best available copy.

PROGRAM	ROFF
18 IF (FKSW) GO TO 23	ROFF 659
NFOOT=NFOOT+1	ROFF 660
CALL NUMBER (LBRCT,NFOOT,RBRCT,INFAKE,I,J,.NOT.FILLSW)	ROFF 661
GO TO 9	ROFF 662
3 UNDERSCORE COMES HERE	ROFF 663
19 USW=.NOT.USW	ROFF 664
PRU=.TRUE.	ROFF 665
I=I+1	ROFF 666
GO TO 9	ROFF 667
3 \$ FORCE SMALL LETTER	ROFF 668
20 CAPSW=.FALSE.	ROFF 669
ALLCAP=.FALSE.	ROFF 670
I=I+1	ROFF 671
GO TO 9	ROFF 672
21 ALLCAP=.NOT.ALLCAP	ROFF 673
I=I+1	ROFF 674
GO TO 9	ROFF 675
3	ROFF 676
3 AT BACKSPACE AND OVERSTRIKE. ATCTR IS NUMBER SEEN IN THIS STRING	ROFF 677
22 ATCTR=ATCTR+1	ROFF 678
PRU=.TRUE.	ROFF 679
I=I+1	ROFF 680
IF (IN(I).EQ.ATSIGN) GO TO 22	ROFF 681
3 NOT BACKSPACE AND SUBSTITUTE	ROFF 682
3 NOTCTR IS THE NUMBER OF NOT S IN THE STRING	ROFF 683
3 INSERT REFERENCE NUMBER	ROFF 684
3 CODE IN PREVIOUS FOOTNOTE NUMBER	ROFF 685
GO TO 9	ROFF 686
3 INSERT REFERENCE NUMBER	ROFF 687
23 NR=F=NREF+1	ROFF 688
CALL NUMBER (LPAREN,NREF,RPAREN,INFAKE,I,J,.NOT.FILLSW)	ROFF 689
GO TO 9	ROFF 690
3 CODE IN PREVIOUS FOOTNOTE.	ROFF 691
24 IF (FRSW) GO TO 25	ROFF 692
CALL NUMBER (LBRCT,NFOOT-JTEMP+240,RBRCT,INFAKE,I,J,.NOT.FILLSW)	ROFF 693
I=I+1	ROFF 694
GO TO 9	ROFF 695
3 CODE IN PREVIOUSLY DEFERRED F REFERENCE NUMBER	ROFF 696
25 CALL NUMBER (LPAREN,NREF-JTEMP+240,RPAREN,INFAKE,I,J,.NOT.FILLSW)	ROFF 697
I=I+1	ROFF 698
GO TO 9	ROFF 699
3	ROFF 700
3	ROFF 701
3 GET HERE AFTER MAPPING IS DONE.	ROFF 702
3	ROFF 703
3	ROFF 704
26 IF (.NOT.CCSW) GO TO 28	ROFF 705
CC=IN(1)	ROFF 706
IF (CC.GT.2) GO TO 27	ROFF 707
IF (CC.EQ.0) PCC=PLUS	ROFF 708
IF (CC.EQ.1) PCC=LBLANK	ROFF 709
IF (CC.EQ.2) PCC=ZERO	ROFF 710
Z=8	ROFF 711
ILENG=LENMAX+1	ROFF 712
GO TO 35	ROFF 713



PROGRAM		ROFF	
27	IF (CC.EQ.BLANK) CALL WRBLNK (1)	ROFF	714
	GO TO 6	ROFF	715
28	WIPE OUT REMAINTS OF ORIGINAL LINE.	ROFF	716
29	DO 29 K=J,IN-1	ROFF	717
	IN(K)=BLANK	ROFF	718
	Z=4	ROFF	719
	I LENG=J	ROFF	720
	IF (FILLSW.AND.CENTER) GO TO 41	ROFF	721
	IF (RIGHT.AND.FILLSW) GO TO 46	ROFF	722
	IF (FILLSW) GO TO 40	ROFF	723
	IF (LINECT.GT.PAGEL) CALL EJECT	ROFF	724
	IF IN NOFILL MODE, COPY THE LINE OUT INTACT	ROFF	725
	LINECT=LINECT+CC	ROFF	726
	IF (.NOT.SPEL SW) GO TO 30	ROFF	727
	CALL SPELL (INFAKE,I LENG)	ROFF	728
	I LENG=INL2	ROFF	729
30	CALL TRANS (INFAKE,I LENG)	ROFF	730
	CHECK FOR CENTER MODE OR RIGHT MODE	ROFF	731
	IF (.NOT.CENT.R.AND..NOT.RIGHT) GO TO 35	ROFF	732
	J=OLENG	ROFF	733
	NBLANK=J	ROFF	734
31	IF (IN(J).NE.BLANK) GO TO 32	ROFF	735
	J=J-1	ROFF	736
	NBLANK=NBLANK+1	ROFF	737
	IF (J) 35,35,31	ROFF	738
32	IF (CENTER) NBLANK=NBLANK/2	ROFF	739
	IF (NBLANK.LT.0) GO TO 35	ROFF	740
	NEND=OLENG-NBLANK	ROFF	741
	DO 33 IJ=1,NEND	ROFF	742
	POS=OLENG-IJ+1	ROFF	743
	IPP1=POS-NBLANK	ROFF	744
33	IN(POS)=IN(IPP1)	ROFF	745
	DO 34 IJ=1,NBLANK	ROFF	746
34	IN(IJ)=BLANK	ROFF	747
	RIGHT=.FALSE.	ROFF	748
	CENTER=.FALSE.	ROFF	749
35	CONTINUE	ROFF	750
	IF (.NOT.CCSW) GO TO 36	ROFF	751
	IF (LINECT.GT.PAGEL.AND.CC.NE.0) CALL EJECT	ROFF	752
	PATCH UP OVERSHOOTS OF INPUT BUFFER	ROFF	753
	LINECT=LINECT+CC	ROFF	754
36	I LEN4=4*I LENG	ROFF	755
	IF (U.NE.6) I LEN4=MAX0(I LEN4,4*LENMAX)	ROFF	756
	I LEN1=I LEN4/4	ROFF	757
	Z4=Z/4	ROFF	758
	IF (U.NE.6) WRITE (U) POS,(IN(I),I=Z4,I LEN1)	ROFF	759
	IF (U.EJ.6) CALL MICRO (POS,IN(Z4),I LEN1-Z4+1)	ROFF	760
	NREC=NREC+1	ROFF	761
	IF (I LENG.LT.31) GO TO 38	ROFF	762
	DO 37 I=81,I LENG	ROFF	763
37	IN(I)=BLANK	ROFF	764
38	IF (.NOT.PKW) GO TO 49	ROFF	765
	IF (U.NE.6) WRITE (U) PLUS,(JLINE(I),I=Z4,I LEN1)	ROFF	766
	IF (U.EJ.6) CALL MICRO (PLUS,ULINE(Z4),I LEN1-Z4+1)	ROFF	767
	NREC=NREC+1	ROFF	768

AFWL-TR-72-139

	PROGRAM	ROFF
39	DO 39 I=1, (LENG ULINE(I)=BLANK CONTINUE PRU=.FALSE. GO TO 49	ROFF 769 ROFF 770 ROFF 771 ROFF 772 ROFF 773
3	COME HERE IF IN FILL MODE	ROFF 774 ROFF 775
40	CONTINUE CALL FILL GO TO 2	ROFF 776 ROFF 777 ROFF 778 ROFF 779 ROFF 780
3	COME HERE IF IN CENTER MODE.	ROFF 781 ROFF 782 ROFF 783 ROFF 784 ROFF 785
41	CALL FILL CENTER=.FALSE. BM1=BUFFL-1 CEN=(OLENG-BM1)/2	ROFF 786 ROFF 787 ROFF 788 ROFF 789 ROFF 790
42	CONTINUE DO 43 KK=1, BM1 K=BUFFL-KK KCEN=K+CEN OUT(KCEN)=OUT(K) IF (OVERSM) OVLIN(KCEN)=OVLIN(K)	ROFF 791 ROFF 792 ROFF 793 ROFF 794 ROFF 795 ROFF 796 ROFF 797 ROFF 798 ROFF 799 ROFF 800 ROFF 801 ROFF 802 ROFF 803 ROFF 804 ROFF 805 ROFF 806 ROFF 807 ROFF 808 ROFF 809 ROFF 810 ROFF 811 ROFF 812 ROFF 813 ROFF 814 ROFF 815 ROFF 816 ROFF 817 ROFF 818 ROFF 819 ROFF 820 ROFF 821 ROFF 822 ROFF 823
43	CONTINUE IF (CEN.EQ.0) GO TO 45 DO 44 K=1, CEN OUT(K)=BLANK IF (OVERSM) OVLIN(K)=BLANK	
44	CONTINUE BUFFL=BUFFL+CEN-1	
45	CALL FLUSH GO TO 6	
3	COME HERE IF IN RIGHT ADJUST MODE FOR THE LINE	
46	CALL FILL RIGHT=.FALSE. BM1=BUFFL-1 CEN=OLENG-BM1 GO TO 42	
3	C	
3	CONTROL WORD DECODER	
47	LZ=LMIN ASV=ADSM FSV=FILLSM CCSV=CC PCCSV=PCC CC=1 PCC=LBLANK	

PROGRAM ROFF

AOSW=.FALSE.	ROFF	824
FILLSW=.FALSE.	ROFF	825
LINS=LMAX-LMIN+1	ROFF	826
IF (LINECT+LINS.GT.PAGE1) CALL EJECT	ROFF	827
INHOLD=IN(2)	ROFF	828
JHOLD=IN(3)	ROFF	829
IF (IN(4).EQ.BLANK.AND.IN(5).EQ.BLANK.AND.IN(6).EQ.BLANK.AND.IN(7)	ROFF	830
1.EQ.BLANK) GO TO 51	ROFF	831
DO 48 I=4,80	ROFF	832
INHOLD(I)=IN(I)	ROFF	833
HOLDSW=.TRUE.	ROFF	834
GO TO 53	ROFF	835
IF (.NOT.EQSW) GO TO 6	ROFF	836
IF (LZ.GE.LMAX) GO TO 52	ROFF	837
LZ=LZ+1	ROFF	838
DO 51 I=1,INLENG	ROFF	839
IN(I)=EQU(I,LZ)	ROFF	840
GO TO 8	ROFF	841
INLENG=80	ROFF	842
LMAX=3	ROFF	843
LMIN=3	ROFF	844
EQSW=.FALSE.	ROFF	845
AOSW=ASV	ROFF	846
FILLSW=FSV	ROFF	847
CC=CCSV	ROFF	848
PCC=PCCSV	ROFF	849
ITEMP=INHOLD	ROFF	850
JTEMP=JHOLD	ROFF	851
IF (ITEMP.EQ..ETTR.E.AND.JTEMP.EQ.LETTRQ) GO TO 53	ROFF	852
D=3-CC	ROFF	853
CALL WRBLNK (D)	ROFF	854
IF (.NOT.HOLDSW) GO TO 56	ROFF	855
DO 54 I=4,80	ROFF	856
IN(I)=INHOLD(I)	ROFF	857
HOLDSW=.FALSE.	ROFF	858
GO TO 56	ROFF	859
	ROFF	860
	ROFF	861
ITEMP=IN(2)	ROFF	862
JTEMP=IN(3)	ROFF	863
IF (ITEMP.EQ..ETTR.E.AND.JTEMP.EQ.LETTRQ.AND.NQSW) GO TO 117	ROFF	864
IF (ITEMP.EQ..ETTR.E.AND.JTEMP.EQ.LETTRQ) CALL EQROFF	ROFF	865
IF (EQSW.AND..NOT.VQSW) GO TO 47	ROFF	866
EQSW=.FALSE.	ROFF	867
	ROFF	868
IF (ITEMP.EQ..ETTR.E.AND.JTEMP.EQ.LETTRQ) GO TO 6	ROFF	869
.BR	ROFF	870
IF (ITEMP.EQ..ETTR.B.AND.JTEMP.EQ.LETTRR) GO TO 60	ROFF	871
.PP	ROFF	872
IF (ITEMP.EQ..ETTR.P.AND.JTEMP.EQ.LETTRP) GO TO 79	ROFF	873
.SP	ROFF	874
IF (ITEMP.EQ..ETTR.S.AND.JTEMP.EQ.LETTRP) GO TO 58	ROFF	875
.BP	ROFF	876
IF (ITEMP.EQ..ETTR.B.AND.JTEMP.EQ.LETTRP) GO TO 59	ROFF	877
.FI	ROFF	878

PROGRAM ROFF

IF (ITEMP.EQ.LETTRF.AND.JTEMP.EQ.LETTRI) GO TO 67	ROFF	879
.NF	ROFF	880
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRF) GO TO 68	ROFF	881
.AD	ROFF	882
IF (ITEMP.EQ.LETTRA.AND.JTEMP.EQ.LETTRD) GO TO 69	ROFF	883
.NJ	ROFF	884
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRJ) GO TO 70	ROFF	885
.LL	ROFF	886
IF (ITEMP.EQ.LETTRL.AND.JTEMP.EQ.LETTRL) GO TO 71	ROFF	887
.CO	ROFF	888
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRC) GO TO 61	ROFF	889
.MA	ROFF	890
IF (ITEMP.EQ.LETTRM.AND.JTEMP.EQ.LETTRA) GO TO 62	ROFF	891
.DS	ROFF	892
IF (ITEMP.EQ.LETTRD.AND.JTEMP.EQ.LETTRS) GO TO 63	ROFF	893
.SS	ROFF	894
IF (ITEMP.EQ.LETTRS.AND.JTEMP.EQ.LETTRS) GO TO 64	ROFF	895
.SK	ROFF	896
IF (ITEMP.EQ.LETTRS.AND.JTEMP.EQ.LETTRK) GO TO 72	ROFF	897
.NE	ROFF	898
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRE) GO TO 73	ROFF	899
.RF	ROFF	900
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRF) GO TO 102	ROFF	901
.RE	ROFF	902
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRE) GO TO 104	ROFF	903
.RP	ROFF	904
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRP) GO TO 106	ROFF	905
.PA	ROFF	906
IF (ITEMP.EQ.LETTRP.AND.JTEMP.EQ.LETTRA) GO TO 74	ROFF	907
.PH	ROFF	908
IF (ITEMP.EQ.LETTRP.AND.JTEMP.EQ.LETTRN) GO TO 76	ROFF	909
.FN	ROFF	910
IF (ITEMP.EQ.LETTRF.AND.JTEMP.EQ.LETTRN) GO TO 90	ROFF	911
.FE	ROFF	912
IF (ITEMP.EQ.LETTRF.AND.JTEMP.EQ.LETTRE) GO TO 94	ROFF	913
.IN	ROFF	914
IF (ITEMP.EQ.LETTRI.AND.JTEMP.EQ.LETTRN) GO TO 77	ROFF	915
.TR	ROFF	916
IF (ITEMP.EQ.LETTRI.AND.JTEMP.EQ.LETTRR) GO TO 83	ROFF	917
.CL	ROFF	918
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRE) GO TO 84	ROFF	919
.RT	ROFF	920
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRI) GO TO 85	ROFF	921
.CH	ROFF	922
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRH) GO TO 86	ROFF	923
.UN UNDET	ROFF	924
IF (ITEMP.EQ.LETTRJ.AND.JTEMP.EQ.LETTRN) GO TO 98	ROFF	925
.NC	ROFF	926
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRC) GO TO 87	ROFF	927
.NS	ROFF	928
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRS) GO TO 88	ROFF	929
.HE	ROFF	930
IF (ITEMP.EQ.LETTRH.AND.JTEMP.EQ.LETTRE) GO TO 123	ROFF	931
.DU	ROFF	932
IF (ITEMP.EQ.LETTRP.AND.JTEMP.EQ.LETTRU) GO TO 118	ROFF	933

PROGRAM	ROFF	
3 .NP	ROFF	934
IF (ITEMP.EQ.LETTRN.AND.JTEMP.EQ.LETTRP) GO TO 119	ROFF	935
3 .CC	ROFF	936
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRC) GO TO 120	ROFF	937
3 .CX	ROFF	938
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRA) GO TO 121	ROFF	939
3 .RA	ROFF	940
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRA) GO TO 89	ROFF	941
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRC) GO TO 114	ROFF	942
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRQ) GO TO 115	ROFF	943
3 .FS SET FOOTNOTE SPACING TO CURRENT VALUE	ROFF	944
IF (ITEMP.EQ.LETTRF.AND.JTEMP.EQ.LETTRS) GO TO 99	ROFF	945
3 .RS SET REFERENCE SPACING TO CURRENT VALUE	ROFF	946
IF (ITEMP.EQ.LETTRR.AND.JTEMP.EQ.LETTRS) GO TO 101	ROFF	947
3 .SF	ROFF	948
IF (ITEMP.EQ.LETTRS.AND.JTEMP.EQ.LETTRF) GO TO 113	ROFF	949
3 .PL SET PAGE LENGTH DEFAULT 48 LINES	ROFF	950
IF (ITEMP.EQ.LETTRP.AND.JTEMP.EQ.LETTRL) GO TO 100	ROFF	951
3 .CT NUMBER FOOTNOTES AND CONTINUOUSLY	ROFF	952
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRT) GO TO 112	ROFF	953
3 .AL	ROFF	954
IF (ITEMP.EQ.LETTRA.AND.JTEMP.EQ.LETTRL) GO TO 116	ROFF	955
3 .FR	ROFF	956
IF (ITEMP.EQ.LETTRF.AND.JTEMP.EQ.LETTRR) GO TO 122	ROFF	957
3 .EF END OF FILE	ROFF	958
IF (ITEMP.EQ.LETTRC.AND.JTEMP.EQ.LETTRF) GO TO 130	ROFF	959
3 .# TURN UNDERSCORE SW ON	ROFF	960
IF (ITEMP.EQ.JSCORE) GO TO 55	ROFF	961
3 .CFX TURN ALL CAP ON	ROFF	962
IF (ITEMP.EQ.FLEX) GO TO 65	ROFF	963
3 UNKNOWN CONTROL WORD. REMEMBER	ROFF	964
57 BADCTR=BADCTR+1	ROFF	965
CALL DISPLA(1,H**ERROR AT ,N)	ROFF	966
GO TO 6	ROFF	967
3 .SP N N BLANK LINES INSERTED	ROFF	968
58 IF (FILLSW) CALL FLUSH	ROFF	969
CALL WRBLNK (INTEG(IN,4,1))	ROFF	970
GO TO 6	ROFF	971
3 .BP BEGIN A NEW PAGE	ROFF	972
59 IF (FILLSW) CALL FLUSH	ROFF	973
CALL EJECT	ROFF	974
CAPSW=.TRUE.	ROFF	975
GO TO 6	ROFF	976
3 .BR BREAK AND START A NEW LINE	ROFF	977
3 FOR NOW , SET CAPSW	ROFF	978
60 CAPSW=.TRUE.	ROFF	979
IF (FILLSW) CALL FLUSH	ROFF	980
GO TO 6	ROFF	981
3 .CO SET COPY SWITCH ON. COPY INPUT, NO MAPPING	ROFF	982
61 COPYSW=.TRUE.	ROFF	983
IF (FILLSW) CALL FLUSH	ROFF	984
J=81	ROFF	985
GO TO 6	ROFF	986
3 .MA MAP, SO SWITCH OFF	ROFF	987
62 COPYSW=.FALSE.	ROFF	988

PROGRAM	ROFF
GO TO 6	ROFF 989
3 .DS DOUBLE SPACE	ROFF 990
53 IF (FILLSW) CALL FLUSH	ROFF 991
CC=2	ROFF 992
PCC=ZERO	ROFF 993
GO TO 6	ROFF 994
3 .SS SINGLE SPACE MODE	ROFF 995
54 IF (FILLSW) CALL FLUSH	ROFF 996
CC=1	ROFF 997
PCC=LBLANK	ROFF 998
GO TO 6	ROFF 999
55 USW=.TRUE.	ROFF 1000
PRU=.TRUE.	ROFF 1001
GO TO 6	ROFF 1002
3 .CFX TURN ON FOR NEXT LINE	ROFF 1003
56 ALLCAP=.TRUE.	ROFF 1004
GO TO 6	ROFF 1005
3 .FI ENTER FILL MODE	ROFF 1006
57 FILLSH=.TRUE.	ROFF 1007
3 ADSW=.TRUE.	ROFF 1008
GO TO 6	ROFF 1009
3 .NF ENTER NOFILL	ROFF 1010
58 FILLSH=.FALSE.	ROFF 1011
ADSW=.FALSE.	ROFF 1012
CALL FLUSH	ROFF 1013
GO TO 6	ROFF 1014
3 .AD TURN ON RIGHT ADJUST MODE	ROFF 1015
59 IF (FILLSW) CALL FLUSH	ROFF 1016
ADSW=.TRUE.	ROFF 1017
FILLSW=.TRUE.	ROFF 1018
GO TO 6	ROFF 1019
3 .NJ TURN OFF RIGHT ADJUST MODE	ROFF 1020
70 IF (FILLSW) CALL FLUSH	ROFF 1021
ADSW=.FALSE.	ROFF 1022
FILLSW=.FALSE.	ROFF 1023
GO TO 6	ROFF 1024
3 .LL SET LINE LENGTH	ROFF 1025
71 IF (FILLSW) CALL FLUSH	ROFF 1026
OLENG=INTEG(IN,4,60)	ROFF 1027
LENMAX=MAX0(LENMAX,OLENG)	ROFF 1028
GO TO 6	ROFF 1029
3 .SK N LEAVE N BLANK PAGES AT THE NEXT OPPORTUNITY	ROFF 1030
3 ADDITIVE ON N UNTIL EXECJTED.	ROFF 1031
72 PAGES=INTEG(IN,4,1)	ROFF 1032
CALL SKIP	ROFF 1033
GO TO 6	ROFF 1034
3 .NE N SKIP TO PAGE IF THERE AREN T AT LEAST N LINES	ROFF 1035
3 ON CURRENT PAGE.	ROFF 1036
73 LINS=INTEG(IN,4,0)	ROFF 1037
IF (LINECT+LINS.LE.PAGE1) GO TO 75	ROFF 1038
IF (FILLSW) CALL FLUSH	ROFF 1039
CALL EJECT	ROFF 1040
GO TO 6	ROFF 1041
3 .PA START A NEW PAGE WITH GIVEN NUMBER. DFLT IS 1	ROFF 1042
74 IF (FILLSW) CALL FLUSH	ROFF 1043

Reproduced from  
best available copy.

PROGRAM	ROFF	
PAGENO=INTEG(IN,4,1)		ROFF 1044
CALL EJECT		ROFF 1045
75 CONTINUE		ROFF 1046
GO TO 6		ROFF 1047
3 .PM SET PAGING MODE.		ROFF 1048
76 PH=INTEG(IN,4,1)		ROFF 1049
IF (PH.GT.2) GO TO 57		ROFF 1050
PHUNSW=.FALSE.		ROFF 1051
IF (PH.EQ.0) GO TO 6		ROFF 1052
PHUNSW=.TRUE.		ROFF 1053
IF (PH.EQ.1.AND.RNUMSW.OR.PH.EQ.2.AND..NOT.RNUMSW) PAGENO=1		ROFF 1054
RNUMSW=.FALSE.		ROFF 1055
IF (PH.EQ.1) GO TO 6		ROFF 1056
RNUMSW=.TRUE.		ROFF 1057
GO TO 6		ROFF 1058
3 .IN N INDENT N SPACES. DDFT OS 3		ROFF 1059
3 MOVES OUTPUT TO 0 + 1 PRINT POSITON		ROFF 1060
77 IF (FILLSW) CALL FLUSH		ROFF 1061
INDENT=INTEG(IN,4,0)		ROFF 1062
BUFFL=INDENT		ROFF 1063
IF (INDENT.EQ.0) GO TO 6		ROFF 1064
IF (INDENT.GT.129.OR.INDENT.LT.0) BADCTR=BADCTR+1		ROFF 1065
IF (INDENT.GT.129.OR.INDENT.LT.0) GO TO 5		ROFF 1066
DO 78 I=1,INDENT		ROFF 1067
78 OUT(I)=PERCEN		ROFF 1068
GO TO 6		ROFF 1069
3 .PP N NEW PARAGRAPH, WITH INDENTING		ROFF 1070
3 START FIRST LINE OF PARAGRAPH AT PP+INDENT		ROFF 1071
3 IF N IS NULL USE PREVIOUS VALUE. OTHERWISE, COMPUTE A NEW ONE		ROFF 1072
3 SET CAPSW ON, AS IN BREAK, OR		ROFF 1073
79 IF (FILLSW) CALL FLUSH		ROFF 1074
3 WHAT IS N		ROFF 1075
PPTMP=INTEG(IN,4,-1)		ROFF 1076
3 IF NEG , WAS DEFAULTED		ROFF 1077
IF (PPTMP.LT.0) GO TO 83		ROFF 1078
3 OTHERWISE RECOMPUTE		ROFF 1079
PP=PPTMP		ROFF 1080
3 INSERT BLANKS		ROFF 1081
80 PPTMP=PP+INDENT		ROFF 1082
IF (PPTMP.LE.0) GO TO 82		ROFF 1083
DO 81 I=1,PPTMP		ROFF 1084
81 OUT(I)=PERCEN		ROFF 1085
82 BUFFL=PPTMP		ROFF 1086
CAPSW=.TRUE.		ROFF 1087
GO TO 6		ROFF 1088
3		ROFF 1089
3 .YR C1 TO C2		ROFF 1090
3 ON OUTPUT , CONVERT ALL INSTANCES OF C1 TO C2		ROFF 1091
3		ROFF 1092
83 CALL TR (IN,4)		ROFF 1093
GO TO 6		ROFF 1094
3 .CE CENTER THE LINE		ROFF 1095
84 CENTER=.TRUE.		ROFF 1096
IF (FILLSW) CALL FLUSH		ROFF 1097
CAPSW=.TRUE.		ROFF 1098

AFWL-TR-72-130

	PROGRAM	ROFF
	GO TO 6	
35	.RT REVERT THE TRANSLATE COMMAND	ROFF 1099
	CALL INITR (INFKE)	ROFF 1100
	GO TO 6	ROFF 1101
36	.CH SPELLING MODE . LOOK FOR SPELLING ERRORS	ROFF 1102
	SPELW=.TRUE.	ROFF 1103
	CALL SEARCH (IN,4,INIJ)	ROFF 1104
	IF (INIJ.EQ.2) GO TO 57	ROFF 1105
	GO TO 6	ROFF 1106
37	.NE REVERT SPELLING CHANGES	ROFF 1107
	SPELW=.FALSE.	ROFF 1108
	GO TO 6	ROFF 1109
38	.NS DO NOT SPAVE THE PRINTER CARRIAGE ON OUTPUT	ROFF 1110
	IF (FILLW) CALL FLUSH	ROFF 1111
	PG=PLUS	ROFF 1112
	CC=0	ROFF 1113
	GO TO 6	ROFF 1114
39	.RA RIGHT ADJUST THE NEXT CARD	ROFF 1115
	RIGHT=.TRUE.	ROFF 1116
	IF (FILLW) CALL FLUSH	ROFF 1117
	GO TO 6	ROFF 1118
40	START FOOTNOTE: -- SAVE THE BUFFERS	ROFF 1119
	IF (FRSW) GO TO 102	ROFF 1120
	ARE WE ALREADY IN THE FOOTNOTE MODE	ROFF 1121
	IF (FTING) GO TO 91	ROFF 1122
	ASSIGN 92 TO STATE	ROFF 1123
	GO TO 125	ROFF 1124
41	CALL FLUSH	ROFF 1125
	GO TO 93	ROFF 1126
42	FTING=.TRUE.	ROFF 1127
	LINECT=0	ROFF 1128
	NREC=0	ROFF 1129
	U=4	ROFF 1130
	IF (.NOT.FTNOTE) LINECT=2	ROFF 1131
	FTNOTE=.TRUE.	ROFF 1132
43	SET NEW VALUES	ROFF 1133
	AOSH=.TRUE.	ROFF 1134
	CAPSW=.TRUE.	ROFF 1135
	FILLW=.TRUE.	ROFF 1136
	COPYSW=.FALSE.	ROFF 1137
	CCSW=.FALSE.	ROFF 1138
	PSW=PSA	ROFF 1139
	CC=FCC	ROFF 1140
	PGC=FGC	ROFF 1141
	INDENT=0	ROFF 1142
	SETUP FOOTNOTE: STUFF IN OUTPUT BUFFER	ROFF 1143
	OUT(1)=LBRCT	ROFF 1144
	NFOOTP=NFOOTP+1	ROFF 1145
	BUFFL=1	ROFF 1146
	CALL NUHSE (LBRCT,NFOOTP,RBRCT,OUT,DJH,BUFFL,.TRUE.)	ROFF 1147
	OUT(BUFFL)=PERCEN	ROFF 1148
	NWORD=0	ROFF 1149
	GO TO 6	ROFF 1150
44	END FOOT NOTE: HERE	ROFF 1151
	IF (FRSW) GO TO 104	ROFF 1152
		ROFF 1153



PROGRAM	ROFF	
CALL FLUSH	ROFF	1154
X=LINECT+CSAVE(1)	ROFF	1155
DO WE HAVE SOMETHING IN THE SAVED BUFFER TO PRINT	ROFF	1156
PRMORE=.FALSE.	ROFF	1157
IF (FLAGSV(2).AND.SAVED(263).GT.0) PRMORE=.TRUE.	ROFF	1158
IF (PRMORE) X=X+CSAVE(2)	ROFF	1159
WHERE WILL THE FOOTNOTE PUT US ON THE PAGE	ROFF	1160
IF (X.LE.PAGEL) GO TO 95	ROFF	1161
IF THE I IS THE FIRST FOOTNOTE AND ARE WE AT BOTTOM - FORGET IT	ROFF	1162
IF (PRMORE.AND.NFOOT.EQ.1.AND.X.GT.PAGEL-4) GO TO 97	ROFF	1163
X=MAX(X-PAGEL,0)	ROFF	1164
FTOVER=FTOVER+X	ROFF	1165
LINECT=LINECT-X	ROFF	1166
IF (PRMORE) CSAVE(1)=CSAVE(1)+CSAVE(2)	ROFF	1167
UPDATE FOOTNOTE COUNTERS AND RESTORE OLD BUFFERS	ROFF	1168
95 FTLINZ=FTLINZ+LINECT	ROFF	1169
FTREC=FTREC+NREC	ROFF	1170
LINECT=LINECT+CSAVE(1)	ROFF	1171
ASSIGN 96 TO STATE	ROFF	1172
GO TO 128	ROFF	1173
96 CONTINUE	ROFF	1174
FTING=.FALSE.	ROFF	1175
GO TO 6	ROFF	1176
97 FTNOT=.FALSE.	ROFF	1177
FTLINZ=LINECT	ROFF	1178
LINECT=CSAVE(1)	ROFF	1179
CSAVE(1)=FTLINZ	ROFF	1180
CALL EJECT	ROFF	1181
FTNOT=.TRUE.	ROFF	1182
LINECT=J	ROFF	1183
GO TO 95	ROFF	1184
98 IF (FILLSH) CALL FLUSH	ROFF	1185
BUFFL=MAX(0,INDENT-INTEG(IN,4,INDENT))	ROFF	1186
GO TO 6	ROFF	1187
99 FCC=CC	ROFF	1188
FPCC=PCC	ROFF	1189
GO TO 6	ROFF	1190
100 PAGEL=INTEG(IN,4,48)	ROFF	1191
GO TO 6	ROFF	1192
101 RCC=CC	ROFF	1193
RPCC=PCC	ROFF	1194
GO TO 6	ROFF	1195
START REFERENCE	ROFF	1196
102 IF (REFING) CALL FLUSH	ROFF	1197
IF (.NOT.REFING) NREC=0	ROFF	1198
ASSIGN 103 TO STATE	ROFF	1199
IF (.NOT.REFING) GO TO 125	ROFF	1200
103 CONTINUE	ROFF	1201
CAPSH=.TRUE.	ROFF	1202
COPYSH=.FALSE.	ROFF	1203
CCSH=.FALSE.	ROFF	1204
PSH=PSA	ROFF	1205
ADSH=.TRUE.	ROFF	1206
FILLSH=.TRUE.	ROFF	1207
	ROFF	1208

AFWL-TR-72-139

PROGRAM ROFF

INDENT=J	ROFF	1209
REFING=.TRUE.	ROFF	1210
CC=RCC	ROFF	1211
PLC=RPCC	ROFF	1212
U=3	ROFF	1213
BUFFL=1	ROFF	1214
NREFP=NREFP+1	ROFF	1215
CALL NUMBER (L.PAREN,NREFP,R.PAREN,OUT,DUM,BUFFL,.TRUE.)	ROFF	1216
OUT(BUFFL)=PERGEN	ROFF	1217
NWURO=0	ROFF	1218
GO TO 6	ROFF	1219
3	ROFF	1220
3	ROFF	1221
3	ROFF	1222
104 ASSIGN 105 TO STATE	ROFF	1223
IF (FILLSH) CALL FLUSH	ROFF	1224
REFREC=REFREC+NREC	ROFF	1225
GO TO 128	ROFF	1226
105 LINECT=CSAVE(1)	ROFF	1227
REFING=.FALSE.	ROFF	1228
GO TO 6	ROFF	1229
3 PRINT FOOTNOTES	ROFF	1230
106 IF (REFREC.LE.0) GO TO 6	ROFF	1231
DO 107 K=1,REFREC	ROFF	1232
107 BACKSPACE J	ROFF	1233
IF (FILLSH) CALL FLUSH	ROFF	1234
CALL EJECT	ROFF	1235
3 OUTPUT 1H ,*REFERENCES*	ROFF	1236
IIOUT(1)=LBLANK	ROFF	1237
IIOUT(2)=LETTAK	ROFF	1238
IIOUT(3)=LETTRE	ROFF	1239
IIOUT(4)=LETTKF	ROFF	1240
IIOUT(5)=LETTKE	ROFF	1241
IIOUT(6)=LETTKR	ROFF	1242
IIOUT(7)=LETTRE	ROFF	1243
IIOUT(8)=LETTKN	ROFF	1244
IIOUT(9)=LETTKU	ROFF	1245
IIOUT(10)=LETTRE	ROFF	1246
IIOUT(11)=LETTKS	ROFF	1247
CALL MICRO (IIOUT(1),IIOUT(2),10)	ROFF	1248
3 OUTPUT 1H+, WITH UNDERSCORES	ROFF	1249
IIOUT(1)=PLUS	ROFF	1250
DO 108 I=2,11	ROFF	1251
108 IIOUT(I)=USCORE	ROFF	1252
CALL MICRO (IIOUT(1),IIOUT(2),10)	ROFF	1253
LINECT=LINECT+2	ROFF	1254
LR=LENMAX+1	ROFF	1255
DO 110 K=1,REFREC	ROFF	1256
110 J1 FORMAT(131A1)	ROFF	1257
3 READ(3,10J1) (FLIN(I),I=1,LR)	ROFF	1258
3 READ(3) (FLIN(I),I=2,LR)	ROFF	1259
IF (EOF(3)) 111,109	ROFF	1260
109 CONTINUE	ROFF	1261
CCU=FLIN(1)	ROFF	1262
IF (LINECT.GT.PAGEL.AND.CCU.NE.PLUS) CALL EJECT	ROFF	1263

PROGRAM	ROFF	
CALL MICRO (FLIN(1),FLIN(2),LR-1)	ROFF	1264
LINECT=LINECT+1	ROFF	1265
IF (CCC.EQ.ZERO) LINECT=LINECT+1	ROFF	1266
110 IF (CCC.EQ.P.US) LINECT=LINECT-1	ROFF	1267
111 CALL EJECT	ROFF	1268
REFREC=0	ROFF	1269
NREF=0	ROFF	1270
NREFP=0	ROFF	1271
GO TO 6	ROFF	1272
3	ROFF	1273
SET THE FOOT NUMBER TO A SPECIFIED VALUE	ROFF	1274
112 CTFN=.TRUE.	ROFF	1275
GO TO 6	ROFF	1276
113 NFOOT=INTEG(IN,4,1)	ROFF	1277
NFOOTP=NFOOT	ROFF	1278
GO TO 6	ROFF	1279
114 EOSW=.TRUE.	ROFF	1280
GO TO 6	ROFF	1281
115 NQSW=.TRUE.	ROFF	1282
GO TO 6	ROFF	1283
116 EOSW=.FALSE.	ROFF	1284
NQSW=.FALSE.	ROFF	1285
GO TO 6	ROFF	1286
117 IF (FILLSW) CALL FLUSH	ROFF	1287
EQSW=.TRUE.	ROFF	1288
CALL WRBLNK (4)	ROFF	1289
GO TO 6	ROFF	1290
118 PSW=.TRUE.	ROFF	1291
PSA=.TRUE.	ROFF	1292
PAGEN=PAGENO-1	ROFF	1293
GO TO 6	ROFF	1294
119 PSW=.FALSE.	ROFF	1295
PSA=.FALSE.	ROFF	1296
GO TO 6	ROFF	1297
120 IF (FILLSW) CALL FLUSH	ROFF	1298
IF (CCSW) GO TO 6	ROFF	1299
AOSA=AOSW	ROFF	1300
FISA=FILLSW	ROFF	1301
PSA=PSW	ROFF	1302
CCHOLD=CC	ROFF	1303
CCSW=.TRUE.	ROFF	1304
PSW=.FALSE.	ROFF	1305
COPYSW=.TRUE.	ROFF	1306
ADSW=.FALSE.	ROFF	1307
FILLSW=.FALSE.	ROFF	1308
GO TO 6	ROFF	1309
121 CCSW=.FALSE.	ROFF	1310
ADSW=AOSA	ROFF	1311
FILLSW=FISA	ROFF	1312
PSW=PSA	ROFF	1313
COPYSW=.FALSE.	ROFF	1314
CAPSW=.FALSE.	ROFF	1315
CC=CCHOLD	ROFF	1316
PCC=LBLANK	ROFF	1317
IF (CC.EQ.2) PCC=ZERO	ROFF	1318

AFWL-TR-72-139

PROGRAM	ROFF	
IF (CC.EQ.3) PCC=PLUS		
PAGEN=PAGENO-1		
GO TO 6		
122 CONTINUE	ROFF	1319
FRSH=.TRUE.	ROFF	1320
GO TO 6	ROFF	1321
123 DO 124 I=1,54	ROFF	1322
124 HEAD(I)=IN(I+4)	ROFF	1323
GO TO 6	ROFF	1324
	ROFF	1325
	ROFF	1326
	ROFF	1327
125 CONTINUE	ROFF	1328
SAVE ALL THE CURRENT STUFF IN THE SAVE BUFFER	ROFF	1329
DO 126 I=1,254	ROFF	1330
126 SAVED(I)=SAVE(I)	ROFF	1331
FLAGSV(1)=ADSH	ROFF	1332
FLAGSV(2)=FILLSH	ROFF	1333
FLAGSV(3)=CAPSH	ROFF	1334
FLAGSV(4)=USH	ROFF	1335
FLAGSV(5)=PRU	ROFF	1336
FLAGSV(6)=ALLCAP	ROFF	1337
FLAGSV(7)=COPYSH	ROFF	1338
FLAGSV(8)=CUSH	ROFF	1339
FLAGSV(9)=PSH	ROFF	1340
CSAVE(1)=LINEST	ROFF	1341
CSAVE(2)=CC	ROFF	1342
CSAVE(3)=PCC	ROFF	1343
CSAVE(4)=INDENT	ROFF	1344
IF (.NOT.OVERSH) GO TO STATE, (92,103)	ROFF	1345
DO 127 I=1,13.	ROFF	1346
127 OVLIN(I)=LBANK	ROFF	1347
OVERSH=.FALSE.	ROFF	1348
GO TO STATE, (92,103)	ROFF	1349
RESTORE SAVED BUFFERS	ROFF	1350
128 CONTINUE	ROFF	1351
CC=CSAVE(2)	ROFF	1352
PCC=CSAVE(3)	ROFF	1353
INDENT=CSAVE(4)	ROFF	1354
ADSH=FLAGSV(1)	ROFF	1355
FILLSH=FLAGSV(2)	ROFF	1356
CAPSH=FLAGSV(3)	ROFF	1357
USH=FLAGSV(4)	ROFF	1358
PRU=FLAGSV(5)	ROFF	1359
ALLCAP=FLAGSV(6)	ROFF	1360
COPYSH=FLAGSV(7)	ROFF	1361
CUSH=FLAGSV(8)	ROFF	1362
PSH=FLAGSV(9)	ROFF	1363
U=0	ROFF	1364
DO 129 I=1,254	ROFF	1365
129 SAVE(I)=SAVED(I)	ROFF	1366
GO TO STATE, (96,105)	ROFF	1367
.EF	ROFF	1368
	ROFF	1369
	ROFF	1370
130 IF (FILLSH) CALL FLUSH	ROFF	1371
	ROFF	1372
	ROFF	1373

AFWL-TR-77-139

	PROGRAM	ROFF		
131	CONTINUE		ROFF	1374
	CALL EJECT		ROFF	1375
	IF (FTNOTE) 33 TO 131		ROFF	1376
	IF (BADCTR.GT.) CALL DISPLA(20HNO OF CONTROL ERRORS ,BADCTR)		ROFF	1377
	CALL QUIT (J)		ROFF	1378
	END		ROFF	1379

AFWL-TR-72-130

BLOCK DATA

BLOCK DATA	ROFF	1380
IMPLICIT INTEGER(A-Z)	ROFF	1381
INTEGER BUFF, CC, EQU, FT LINZ, FTOVER, FTREC, O LENG, OUT, OVLINE, PAGEL, PA	ROFF	1382
1 GENO, U, ULINE	ROFF	1383
INTEGER PCC	ROFF	1384
COMMON /EQBUF/ EQU(200,4), LMIN, LMAX, EQSW	ROFF	1385
COMMON /OUTBUF/ OUT(130), OVLINE(130), BUFFL, OVERSH, NWORD, O LENG, PSW,	ROFF	1386
1 LENMAX	ROFF	1387
COMMON /INBUF/ IN(99), JLINE(99), PRU, INLENG, INL1	ROFF	1388
COMMON /UPARM/ CC, PCC, INDENT, PAGENO, INECT, PAGEL, PHONSW, RNUMSW	ROFF	1389
COMMON /FEET/ U, NREC, NFOOT, FTREC, FTNOTE, NFOOTP, FTOVER, FT LINZ, CTFN	ROFF	1390
LOGICAL EQSW, OVERSH, PSW, PRU, PHONSW, RNUMSW, CTFN, FTNOTE	ROFF	1391
DATA OUT/130*54/, OVLINE/130*64/, IN/99*64/, ULINE/99*64/	ROFF	1392
DATA LMAX/3/, LMIN/3/, EQSW/.FALSE./, INLENG/80/, PRU/.FALSE./	ROFF	1393
DATA PSW/.FALSE./, O LENG/60/, LENMAX/60/, BUFFL/0/, NWORD/0/	ROFF	1394
DATA PAGENO/1/, CC/1/, INDENT/0/, PAGEL/42/, INECT/1/, PHONSW/.TRUE./,	ROFF	1395
1 RNUMSW/.FALSE./, OVERSH/.FALSE./	ROFF	1396
DATA NFOOT/0/, NFOOTP/0/, NREC/0/, FT LINZ/0/, FTOVER/0/, FTREC/0/, CTFN/	ROFF	1397
1 .FALSE./, U/6/, FTNOTE/.FALSE./	ROFF	1398
DATA PAGEL/48/	ROFF	1399
END	ROFF	1400

SUBROUTINE EJECT

	SUBROUTINE EJECT	ROFF	1401
3	SUBROUTINE EJECT	ROFF	1402
	INTEGER ONE	ROFF	1403
	INTEGER USCORE, NUMBER(10), IDATA2(42)	ROFF	1404
	INTEGER FTLINZ, FTOVER, FLIN, 3, FTREC, PAGENO, FTOVER, TSKIP	ROFF	1405
	INTEGER BUFL, CC, HEAD, OLENG, OUT, OVLIN, PAGEL, PCC, U	ROFF	1406
	LOGICAL OVERSW, PSM	ROFF	1407
4	IMPLICIT INTEGER (A-Z)	ROFF	1408
	COMMON /OPAR4/ CC, PCC, INDENT, PAGENO, LINECT, PAGEL, PHONSW, RNUMSW	ROFF	1409
	COMMON /FEET/ U, NREC, NFOOT, FTREC, FTNOTE, NFOOTP, FTOVER, FTLINZ, CTFN	ROFF	1410
	DIMENSION IP30(6)	ROFF	1411
	COMMON /OUTBJF/ OUT(130), OVLIN(130), BUFL, OVERSW, NWORD, OLENG, PSM,	ROFF	1412
	LENMAX	ROFF	1413
	COMMON /FLINX/ FLIN(131), HEAD(54), IP50	ROFF	1414
	COMMON /SKIPL/ NSKIP	ROFF	1415
	LOGICAL CTFN	ROFF	1416
	LOGICAL FTNOTE	ROFF	1417
	LOGICAL RNUMSW	ROFF	1418
	LOGICAL PHONSW	ROFF	1419
	INTEGER ZERO, BLANK, PLUS	ROFF	1420
	INTEGER TT, TOPSP	ROFF	1421
	INTEGER Z	ROFF	1422
	DATA ZERO, BLANK, PLUS/1H0,1H ,1H+/	ROFF	1423
	DATA USCORE/1.9/	ROFF	1424
	DATA IBLNK/64/	ROFF	1425
	DATA NUMBER/2,C,241,242,243,244,245,246,247,248,249/	ROFF	1426
	DATA ONE/1H1/, TOPSP/3/, TSKIP/1/	ROFF	1427
	DATA LETTR1/137/, LETTRV/165/, LETTRX/167/	ROFF	1428
	IF (.NOT.FTNOTE) GO TO 11	ROFF	1429
2	ARE WE ALREADY AT THE BOTTOM OF THE PAGE	ROFF	1430
	LL=PAGEL-LINECT+1	ROFF	1431
	IF (LL.LE.0) GO TO 2	ROFF	1432
	DO 1 I=1,LL	ROFF	1433
	CALL MICRO (BLANK,0,0)	ROFF	1434
1	CONTINUE	ROFF	1435
2	CONTINUE	ROFF	1436
	DO 3 I=1,FTREC	ROFF	1437
3	BACKSPACE 4	ROFF	1438
	IDATA2(1)=BLANK	ROFF	1439
	DO 4 I=2,42	ROFF	1440
4	IDATA2(I)=USCORE	ROFF	1441
	CALL MICRO (IDATA2(1),IDATA2(2),41)	ROFF	1442
	FTLINZ=FTLINZ-2	ROFF	1443
	LR=LENMAX+1	ROFF	1444
	IF (FTOVER.LE.0) GO TO 8	ROFF	1445
5	READ (4) (FLIN(I),I=1,LR)	ROFF	1446
2	WRITE OUT AS MUCH OF FOOTNOTES AS WILL FIT	ROFF	1447
	C=FLIN(1)	ROFF	1448
6	FTREC=FTREC-1	ROFF	1449
	IF (C.EQ.ZERO) FTLINZ=FTLINZ-2	ROFF	1450
	IF (C.EQ.BLANK) FTLINZ=FTLINZ-1	ROFF	1451
2	WRITE(6,2000) (FLIN(I),I=1,LR)	ROFF	1452
	CALL MICRO (FLIN(1),FLIN(2),LR-1)	ROFF	1453
	IF (FTLINZ.GT.0) GO TO 5	ROFF	1454
2	HAVE WE PRINTED THE LAST LINE	ROFF	1455

## SUBROUTINE EJECT

	IF (FTREC.EQ.0) GO TO 10	ROFF	1456
2	MOVE TO THE END OF THE DATA SET	ROFF	1457
	DO 7 Z=1,FTREC	ROFF	1458
	READ (4) (FLIN(I),I=1,LR)	ROFF	1459
	C=FLIN(1)	ROFF	1460
	IF (C.EQ.PLUS.AND.Z.EQ.1) GO TO 6	ROFF	1461
7	CONTINUE	ROFF	1462
8	WRITE OUT ALL THE FOOTNOTES AND RESET ALL THE POINTERS	ROFF	1463
	GO TO 11	ROFF	1464
9	DO 9 Z=1,FTREC	ROFF	1465
	READ (4) (FLIN(I),I=1,LR)	ROFF	1466
	CALL MICRO (FLIN(1),FLIN(2),LR-1)	ROFF	1467
9	CONTINUE	ROFF	1468
10	REWIND 4	ROFF	1469
	FTLINZ=3	ROFF	1470
	FTNOTE=.FALSE.	ROFF	1471
	FTREC=0	ROFF	1472
10	ARE WE NUMBERING CONTINUOUSLY	ROFF	1473
	IF (CTFN) GO TO 11	ROFF	1474
	NFCOTP=3	ROFF	1475
	NFOOT=0	ROFF	1476
11	CONTINUE	ROFF	1477
11	IF PM OFF, SKIP, PRINT TOPSP+1 LINES	ROFF	1478
11	IF PM ON, SKIP, PRINT PAGENO, TOPSP LINES	ROFF	1479
	DO 25 J=1,TSKIP	ROFF	1480
	CALL MICRO (JVE,J,L)	ROFF	1481
	IF (PMUNSW) GO TO 12	ROFF	1482
	TT=TOPSP+1	ROFF	1483
	GO TO 23	ROFF	1484
12	CONTINUE	ROFF	1485
12	BLANK OUT OUTPUT	ROFF	1486
	DO 13 I=1,5	ROFF	1487
13	IPGO(I)=IBLN	ROFF	1488
	IF (RNUMSW) GO TO 15	ROFF	1489
	DO 14 L=1,c	ROFF	1490
	N=MOD(PAGENO/10**((L-1),13)+1	ROFF	1491
	IPGO(7-L)=NUMBER(N)	ROFF	1492
	IF (PAGENO.LT.10**L) GO TO 22	ROFF	1493
14	CONTINUE	ROFF	1494
14	ROMAN NUMBER	ROFF	1495
15	IP6=MIN(PAGENO,20)	ROFF	1496
	N=MOD(IP6,5)	ROFF	1497
	IF (N.EQ.0) GO TO 20	ROFF	1498
	GO TO (18,17,16,19), N	ROFF	1499
16	IPGO(4)=LETTRI	ROFF	1500
17	IPGO(5)=LETTRI	ROFF	1501
18	IPGO(6)=LETTRI	ROFF	1502
	NEXT=c-N	ROFF	1503
	GO TO 21	ROFF	1504
19	IPGO(5)=LETTRI	ROFF	1505
	IPGO(6)=LETTX	ROFF	1506
	IF (((IP6/5)/2)*2.EQ.(IP6/5)) IPGO(6)=LETTV	ROFF	1507
	NEXT=4	ROFF	1508
	GO TO 21	ROFF	1509
20	IPGO(6)=LETTV	ROFF	1510



AFWL-TR-72-139

	SUBROUTINE EJECT		
	IF (((IP6/5)/2)*2.EQ.(IP6/5)) IPGO(6)=LETTRX	ROFF	1511
	NEXT=5	ROFF	1512
21	IF (IP6.GT.10) IPGO(NEXT)=LETTRX	ROFF	1513
22	CONTINUE	ROFF	1514
	CALL MICRO (BLANK,HEAD,63)	ROFF	1515
	TT=TOPSP	ROFF	1516
23	IF (CC.EQ.2) TT=TT-1	ROFF	1517
	DO 24 I=1,TT	ROFF	1518
24	CALL MICRO (BLANK,C,0)	ROFF	1519
	PAGENO=PAGENO+1	ROFF	1520
25	CONTINUE	ROFF	1521
	TSKIP=1	ROFF	1522
	LINECT=1	ROFF	1523
	IF (.NOT.FTNOTE) RETURN	ROFF	1524
	SETUP THE OVERFLOW OF THE FOOTNOTE	ROFF	1525
	DID WE RUN PAST THE LAST PAGE BOTTOM BY ONE LINE	ROFF	1526
	FTOVER=FTOVER+FTLINZ	ROFF	1527
	LINECT=MOD(3+FTOVER,PAGE1)	ROFF	1528
	FTOVER=MAX(0,FTOVER-LINECT+3)	ROFF	1529
	FTLINZ=LINECT-1	ROFF	1530
	RETURN	ROFF	1531
		ROFF	1532
	ENTRY SKIP(NSKIP)	ROFF	1533
	ENTRY SKIP	ROFF	1534
	TSKIP=TSKIP+NSKIP	ROFF	1535
	RETURN	ROFF	1536
	END	ROFF	1537
		ROFF	1538

SUBROUTINE FLUSH

	SUBROUTINE FLUSH	ROFF	1539
	INTEGER BB,3,ANK,BUFL,CC,JUT,OVLIN,PAGEL,PCC,PERCEN,PLUS,U	ROFF	1540
	INTEGER CTFN,FTLINZ,FTOVER,FTREC,OLENG,PAGENO	ROFF	1541
	LOGICAL RNUMSH	ROFF	1542
3	IMPLICIT INTEGER (A-Z)	ROFF	1543
	COMMON /OPARY/ CC,PCC,INDENT,PAGENO,LINECT,PAGEL,PHONSH,RNUMSH	ROFF	1544
	COMMON /FELT/ U,NREC,NFOOT,FTREC,FTNOTE,NFOOTP,FTOVER,FTLINZ,CTFN	ROFF	1545
	LOGICAL FTNOTE	ROFF	1546
	LOGICAL PHONSH	ROFF	1547
	COMMON /OUTDJF/ OUT(130),OVLIN(130),BUFL,OVERSH,NWORD,OLENG,PSH,	ROFF	1548
	LENMAX	ROFF	1549
3	WRITE OUTPUT BUFFER	ROFF	1550
	LOGICAL OVERSH	ROFF	1551
	LOGICAL PSH	ROFF	1552
	DATA PLUS/1H+/	ROFF	1553
	DATA PERCEN/136/	ROFF	1554
	DATA BLANK/64/	ROFF	1555
	IF (BUFL.EQ.INDENT.AND.NWORD.EQ.0) RETURN	ROFF	1556
	IF (LINECT.GT.PAGEL.AND.CC.ST.G.AND.J.EQ.6) CALL EJECT	ROFF	1557
	CALL TRANS (JUT,BUFL)	ROFF	1558
	IF (U.EQ.6.OV.BUFL.GE.LENMAX) GO TO 2	ROFF	1559
	BB=BUFL+1	ROFF	1560
	DO 1 JJ=BB,LENMAX	ROFF	1561
1	OUT(JJ)=BLANK	ROFF	1562
	BUFL=LENMAX	ROFF	1563
2	CONTINUE	ROFF	1564
	IF (U.NE.6) WRITE (U) PCC,(OUT(I),I=1,BUFL)	ROFF	1565
	IF (U.EQ.6) CALL MICRO (PCC,OUT,BUFL)	ROFF	1566
	NREC=NREC+1	ROFF	1567
	IF (.NOT.OVERSH) GO TO 4	ROFF	1568
	OVERSH=.FALSE.	ROFF	1569
	IF (U.NE.6) WRITE (U) PLUS,(OVLIN(I),I=1,BUFL)	ROFF	1570
	IF (U.EQ.6) CALL MICRO (PLUS,OVLIN,BUFL)	ROFF	1571
	NREC=NREC+1	ROFF	1572
	DO 3 I=1,BUFL	ROFF	1573
3	OVLIN(I)=BLANK	ROFF	1574
4	CONTINUE	ROFF	1575
	BUFL=INDENT	ROFF	1576
	LINECT=LINECT+CC	ROFF	1577
	NWORD=0	ROFF	1578
	IF (INDENT.LE.6) RETURN	ROFF	1579
	DO 5 I=1,INDENT	ROFF	1580
5	OUT(I)=PERCEN	ROFF	1581
	RETURN	ROFF	1582
	END	ROFF	1583

SUBROUTINE WRBLNK

	SUBROUTINE WRBLNK (N)	ROFF	1584
	INTEGER OUT,OVLINE,PAGEL,U,WHERE	ROFF	1585
	INTEGER BUFF,CC,CTFN,FTLINZ,FTOVER,FTREC,OLENG,PAGENO,PCC	ROFF	1586
	LOGICAL OVERSW,RNUMSW	ROFF	1587
3	IMPLICIT INTEGER (A-Z)	ROFF	1588
	COMMON /OPARM/ CC,PCC,INDENT,PAGENO,LINECT,PAGEL,PHONSW,RNUMSW	ROFF	1589
	COMMON /FEET/ U,NREC,NFOOT,FTREC,FTNOTE,NFOOTP,FTOVER,FTLINZ,CTFN	ROFF	1590
	COMMON /OUTBJ/ OUT(130),OVLINE(130),BUFFL,OVERSW,NWORD,OLENG,PSW,	ROFF	1591
	LENMAX	ROFF	1592
	INTEGER BLANK	ROFF	1593
	LOGICAL PSW	ROFF	1594
	LOGICAL FNOT	ROFF	1595
	LOGICAL PHONSW	ROFF	1596
	DATA BLANK/14 /	ROFF	1597
3	OPERATES IN V-ED MODE -- IF ASK FOR N SPACES, WILL SKIP	ROFF	1598
3	TO NEW PAGE TO GET THEM IF NECESSARY	ROFF	1599
3	SAVE PTR	ROFF	1600
	WHERE=LINECT+V-1	ROFF	1601
3	DOES IT FIT ON CURRENT PAGE	ROFF	1602
	IF (WHERE.LE.PAGEL.OR.U.NE.5) GO TO 1	ROFF	1603
3	NO, SKIP TO NEW PAGE	ROFF	1604
	CALL EJECT	ROFF	1605
3	WERE THERE REALLY ENOUGH SPACES	ROFF	1606
	IF (WHERE.LE.PAGEL+5.AND..NOT.FTNOTE) RETURN	ROFF	1607
3	NO, SO MAKE THEM	ROFF	1608
1	CONTINUE	ROFF	1609
	LINECT=LINECT+N	ROFF	1610
	DO 2 I=1,N	ROFF	1611
	IF (U.EQ.6) CALL MICRO (BLANK,0,0)	ROFF	1612
	IF (U.NE.6) WRITE (U) BLANK	ROFF	1613
	NREC=NREC+1	ROFF	1614
2	CONTINUE	ROFF	1615
	RETURN	ROFF	1616
	END	ROFF	1617

## SUBROUTINE ADJUST

	SUBROUTINE ADJUST	ROFF	1618
	INTEGER BLANK,BUFFL,HOLES,OLENG,OUT,JVLINE,PSH,RSPACE,SUMSIZ	ROFF	1619
3	IMPLICIT INTEGER (A-Z)	ROFF	1620
	COMMON /OUTBUF/ OUT(130),OVLINE(130),BUFFL,OVERSH,NWORD,OLENG,PSH,	ROFF	1621
	1LENMAX	ROFF	1622
	LOGICAL OVERSH	ROFF	1623
	DATA BLANK/64/	ROFF	1624
3	IF ONE WORD, LEAVE	ROFF	1625
	IF (NWORD.LE.1) RETURN	ROFF	1626
3	WHEN ENTER, BUFFL POINTS TO LAST BLANK IN BUFFER.	ROFF	1627
3	WHEN LEAVE, WILL EQUAL OLENG, AND POINTS TO LAST ACTIVE CHARACTER	ROFF	1628
3	NWORD = NUMBER OF ACTUAL WORDS IN LINE	ROFF	1629
3	SUMSIZ = TOTAL SIZE OF ACTIVE WORDS	ROFF	1630
	SUMSIZ=BUFFL-NWORD	ROFF	1631
3	J IS LAST ACTIVE CHAR IN BUF	ROFF	1632
	J=BUFFL-1	ROFF	1633
	HOLES=NWORD-1	ROFF	1634
3	RSPACE IS NUMBER OF SPACES TO BE INSERTED IN TOTAL	ROFF	1635
	RSPACE=OLENG-(SUMSIZ+HOLES)	ROFF	1636
	IF (RSPACE.LE.0) GO TO 5	ROFF	1637
3	BLANK OUT BUFFER	ROFF	1638
	DO 1 I=BUFFL,OLENG	ROFF	1639
	OVLINE(I)=BLANK	ROFF	1640
1	OUT(I)=BLANK	ROFF	1641
3	K IS POINTER IN TARGET	ROFF	1642
	K=OLENG	ROFF	1643
3	MOVE THE CHARS	ROFF	1644
2	IF (OUT(J).EQ.BLANK) GO TO 4	ROFF	1645
	OUT(K)=OUT(J)	ROFF	1646
	OUT(J)=BLANK	ROFF	1647
	IF (.NOT.OVERSH) GO TO 3	ROFF	1648
	OVLINE(K)=OVLINE(J)	ROFF	1649
	OVLINE(J)=BLANK	ROFF	1650
3	CONTINUE	ROFF	1651
	K=K-1	ROFF	1652
	J=J-1	ROFF	1653
	GO TO 2	ROFF	1654
3	WORD IS MOVED. RESET POINTERS TO INSERT BLANKS	ROFF	1655
3	IF SPACES GO EVENLY AMONG HOLES, NO PROB. OTHERWISE, ADD EXTRAS	ROFF	1656
3	NBL IS NUMBER OF EXTRA BLANKS	ROFF	1657
4	NBL=RSPACE/HOLES	ROFF	1658
	IF (RSPACE.NE.NBL*HOLES) NBL=NBL+IRV(1)	ROFF	1659
3	REMAINING SPACES	ROFF	1660
	RSPACE=RSPACE-NBL	ROFF	1661
	IF (RSPACE.LE.0) GO TO 5	ROFF	1662
	HOLES=HOLES-1	ROFF	1663
	K=K-NBL-1	ROFF	1664
	J=J-1	ROFF	1665
	GO TO 2	ROFF	1666
3		ROFF	1667
3	NWORD=0	ROFF	1668
	BUFFL=OLENG	ROFF	1669
	RETURN	ROFF	1670
	END	ROFF	1671

SUBROUTINE NUMBER			
	SUBROUTINE NUMBER (LEFT,N,RIGHT,IN,I,J,COPYSH)	ROFF	1672
3	SUBROUTINE TO INSERT THE NUMBER IN LITERALS SPECIFIED BY N	ROFF	1673
2	SURROUNDED BY THE CHARACTERS SPECIFIED BY LEFT AND RIGHT	ROFF	1674
3	STARTING AT POSITION J AND MOVING THE INPUT LINE TO THE RIGHT	ROFF	1675
3	TO ELIMINATE OVERWRITING.	ROFF	1676
3	IMPLICIT INTEGER (A-Z)	ROFF	1677
	INTEGER ULINE	ROFF	1678
	INTEGER PRU,RIGHT,POW10	ROFF	1679
	COMMON /INBUF/ INN(99),ULINE(99),PRU,INLENG,INL1	ROFF	1680
	COMMON /SPP/ SP	ROFF	1681
	DIMENSION IN(130)	ROFF	1682
3	FINE OUT HOW MANY DIGITS TO WRITE	ROFF	1683
	LOGICAL COPYSH,SP	ROFF	1684
	POW10=0	ROFF	1685
1	POW10=POW10+1	ROFF	1686
	IF (N.GE.10**POW10) GO TO 1	ROFF	1687
3	DO WE HAVE TO MOVE THE INPUT CARD OVER	ROFF	1688
	IMOVE=POW10+1-I+J	ROFF	1689
	IF (SP) IMOVE=IMOVE+1	ROFF	1690
	IF (COPYSH.OR.IMOVE.LE.0) GO TO 4	ROFF	1691
	IF (I.EQ.INL1-1) GO TO 3	ROFF	1692
3	HOW MANY COLUMNS DO WE MOVE	ROFF	1693
	NMOVE=INL1-I-1	ROFF	1694
	DO 2 K=1,NMOVE	ROFF	1695
2	IN(INL1+IMOVE-K)=IN(INL1-K)	ROFF	1696
3	INL1=INL1+IMOVE	ROFF	1697
3	PUT IN THE NUMBERS	ROFF	1698
4	I2=0	ROFF	1699
	DO 5 K=1,POW10	ROFF	1700
	I1=N/10**(POW10-K)	ROFF	1701
	IN(J+K)=240+I1-10*I2	ROFF	1702
5	I2=I1	ROFF	1703
3	PUT IN THE BRACKETS ETC.	ROFF	1704
	IN(I)=LEFT	ROFF	1705
	IN(J+POW10+1)=RIGHT	ROFF	1706
3	UPDATE THE POINTERS	ROFF	1707
	IF (SP) IN(J+POW10+2)=1	ROFF	1708
	I=I+1+MAX0(IMOVE,0)	ROFF	1709
	J=J+POW10+2	ROFF	1710
	IF (SP) J=J+1	ROFF	1711
	SP=.FALSE.	ROFF	1712
	RETURN	ROFF	1713
	END	ROFF	1714

## FUNCTION INTEG

INTEGER FUNCTIONINTEG(IN,START,DEFLT)

PICK UP AN INTEGER IN IN(START) ... IN(82). IF BLANK, RETURN DFL

IMPLICIT INTEGER(A-Z)

INTEGER BLANK,START,DEFLT

DIMENSION IN(82)

DATA BLANK/64/

DO 1 I=START,51

IF (IN(I).NE.BLANK) GO TO 2

CONTINUE

FALL OUT, BLANK, GIVE DEFAULT VALUE

INTEG=DEFLT

RETURN

NORMAL PATH

INTEG=IN(I)-240

I=I+1

JTEMP=IN(I)

IF (JTEMP.EQ.BLANK) RETURN

INTEG=10\*INTEG+(JTEMP-240)

GO TO 3

END

ROFF	1715
ROFF	1716
ROFF	1717
ROFF	1718
ROFF	1719
ROFF	1720
ROFF	1721
ROFF	1722
ROFF	1723
ROFF	1724
ROFF	1725
ROFF	1726
ROFF	1727
ROFF	1728
ROFF	1729
ROFF	1730
ROFF	1731
ROFF	1732
ROFF	1733
ROFF	1734
ROFF	1735

## SUBROUTINE TRANS

3	SUBROUTINE TRANS (BUF,LEN)	ROFF	1736
3	IMPLICIT INTEGER (A-Z)	ROFF	1737
3		ROFF	1738
3	TRANSLATE THE CHARACTERS IN THE OUTPJT BUFFER TO FINAL PRINT FORM	ROFF	1739
3	ACCORDING TO TRTAB	ROFF	1740
3		ROFF	1741
3	DIMENSION IN(81)	ROFF	1742
3	INTEGER TRTAB, BLANK, PERCEN, C1, C2, START	ROFF	1743
3	INTEGER BUF(136)	ROFF	1744
3	COMMON /SR2/ TRTAB(256)	ROFF	1745
3	DATA BLANK, INBLNK, PERCEN/64, 1, 108/	ROFF	1746
3		ROFF	1747
3	DO 1 I=1, L=LEN	ROFF	1748
3	IBUF=BUF(I)	ROFF	1749
3	BUF(I)=TRTAB(IBUF)	ROFF	1750
1	CONTINUE	ROFF	1751
3	RETURN	ROFF	1752
3		ROFF	1753
3		ROFF	1754
3	ENTRY TR	ROFF	1755
3		ROFF	1756
3		ROFF	1757
3	ENTRY TR(IN, START)	ROFF	1758
3		ROFF	1759
3	DO 2 I=1, 81	ROFF	1760
2	IN(I)=BUF(I)	ROFF	1761
3	START=LEN	ROFF	1762
3	EXTRACT C1 AND C2 FROM THE INPUT CONTROL CARD AND PLACE INTO TABL	ROFF	1763
3	DO 3 I=START, 80	ROFF	1764
3	IF (IN(I).NE.BLANK) GO TO 4	ROFF	1765
3	CONTINUE	ROFF	1766
3	IF NO CHARACTERS ON CARD, RETURN	ROFF	1767
3	RETURN	ROFF	1768
3	WE NOW HAVE C1	ROFF	1769
4	C1=IN(I)	ROFF	1770
3	I=I+1	ROFF	1771
3	DO 5 J=I, 81	ROFF	1772
3	IF (IN(J).NE.BLANK) GO TO 6	ROFF	1773
3	CONTINUE	ROFF	1774
3		ROFF	1775
3	C2 WILL BE BLANK IF NOT SPECIFIED	ROFF	1776
3	C2=BLANK	ROFF	1777
3	GO TO 7	ROFF	1778
3	C2=IN(J)	ROFF	1779
7	TRTAB(C1)=C2	ROFF	1780
3	RETURN	ROFF	1781
3		ROFF	1782
3	ENTRY INITTR	ROFF	1783
3	INITIALIZE THE TABLE	ROFF	1784
3		ROFF	1785
3	DO 8 I=1, 255	ROFF	1786
3	TRTAB(I)=I	ROFF	1787
3	CONTINUE	ROFF	1788
3	TRTAB(PERCEN)=BLANK	ROFF	1789
3	TRTAB(INBLNK)=BLANK	ROFF	1790

AFWL-TR-72-139

SUBROUTINE TRANS

RETURN  
END

ROFF 1791  
ROFF 1792



AFWL-TR-72-139

SUBROUTINE SEARCH

2	SUBROUTINE SEARCH (IN, START1, INIJ)	ROFF	1793
2	SUBROUTINE SEARCH (IN, START1, *)	ROFF	1794
	IMPLICIT INTEGER(A-Z)	ROFF	1795
	INTEGER START1	ROFF	1796
	INTEGER ADD, J, ANK, COLUMN, COR, CORECT, COR1, COR2, END, SIGNAL, START	ROFF	1797
	COMMON /SR3/ ITAB(256)	ROFF	1798
	DIMENSION CORECT(22, 20), IN(1)	ROFF	1799
	COMMON /SR/ COLUMN, INL2	ROFF	1800
	DATA BLANK/6+/	ROFF	1801
	INIJ=0	ROFF	1802
	IF (COLUMN.GE.20) INIJ=2	ROFF	1803
	IF (COLUMN.GE.20) RETURN	ROFF	1804
	IF (COLUMN.GE.20) RETURN	ROFF	1805
	START=START1	ROFF	1806
	DO 1 I=START, 90	ROFF	1807
	IF (IN(I).NE.BLANK) GO TO 2	ROFF	1808
1	CONTINUE	ROFF	1809
2	NO STRINGS WERE LOCATED	ROFF	1810
2	RETURN	ROFF	1811
2	SIGNAL=IN(I)	ROFF	1812
	START=I+1	ROFF	1813
	IF (START.GE.80) RETURN	ROFF	1814
3	INITIALIZE WORD LENGTHS	ROFF	1815
	LENG1=0	ROFF	1816
	LENG2=0	ROFF	1817
	END=START+10	ROFF	1818
	DO 3 I=START, END	ROFF	1819
	IF (IN(I).EQ.SIGNAL) GO TO 4	ROFF	1820
	LENG1=LENG1+1	ROFF	1821
3	CORECT (LENG1+2, COLUMN+1)=IN(I)	ROFF	1822
	INIJ=2	ROFF	1823
	RETURN	ROFF	1824
2	RETURN	ROFF	1825
4	CORECT (1, COLUMN+1)=LENG1	ROFF	1826
	START=I+1	ROFF	1827
	END=START+9	ROFF	1828
	DO 5 I=START, END	ROFF	1829
	LENG2=LENG2+1	ROFF	1830
	IF (IN(I).EQ.SIGNAL) GO TO 5	ROFF	1831
5	CORECT (LENG2+12, COLUMN+1)=IN(I)	ROFF	1832
	IF (IN(END+1).EQ.SIGNAL) GO TO 6	ROFF	1833
	INIJ=2	ROFF	1834
	RETURN	ROFF	1835
2	RETURN	ROFF	1836
2	CORECT (2, COLUMN+1)=LENG2	ROFF	1837
	COLUMN=COLUMN+1	ROFF	1838
	RETURN	ROFF	1839
3	*****	ROFF	1840
2	ENTRY SPELL	ROFF	1841
2	*****	ROFF	1842
	INL1=START1	ROFF	1843
2	ENTRY SPELL (IN, INL1)	ROFF	1844
	IF (COLUMN.EQ.0) RETURN	ROFF	1845
		ROFF	1846
		ROFF	1847

## SUBROUTINE SEARCH

	INL2=INL1	ROFF	1848
3	LOOP FOR EACH WORD TO BE CHECKED	ROFF	1849
	DO 15 I=1,COLUMN	ROFF	1850
	END=INL2+1-CORRECT(1,I)	ROFF	1851
	J=0	ROFF	1852
3	LOOK FOR FIRST LETTER	ROFF	1853
7	J=J+1	ROFF	1854
	IF (J.GT.END) GO TO 15	ROFF	1855
	IF (IN(J).NE.CORRECT(J,I)) GO TO 7	ROFF	1856
3	CHECK FOR REST OF WORD	ROFF	1857
	COR=CORRECT(1,I)+J-1	ROFF	1858
	DO 8 K=J,COR	ROFF	1859
	IF (IN(K).NE.CORRECT(K+3-J,I)) GO TO 7	ROFF	1860
9	CONTINUE	ROFF	1861
3	WHICH WAY DO WE MOVE THE REST OF THE CARD	ROFF	1862
	ADD=CORRECT(2,I)-CORRECT(1,I)	ROFF	1863
	IF (ADD.EQ.0) GO TO 13	ROFF	1864
3	MOVE LEFT	ROFF	1865
	COR1=COR+1	ROFF	1866
	IF (ADD.GT.0) GO TO 10	ROFF	1867
	DO 9 K=COR1,INL2	ROFF	1868
	KADD=K+ADD	ROFF	1869
3	IN(KADD)=IN(K)	ROFF	1870
3	11+ IN(K+ADD)=IN(K)	ROFF	1871
	GO TO 12	ROFF	1872
3	MOVE RIGHT	ROFF	1873
10	CONTINUE	ROFF	1874
	DO 11 K=COR1,INL2	ROFF	1875
	KK=INL2+COR1-K	ROFF	1876
11	IN(KKADD)=IN(KK)	ROFF	1877
	KKADD=KK+ADD	ROFF	1878
3	11+ IN(KK+ADD)=IN(KK)	ROFF	1879
12	CONTINUE	ROFF	1880
	INL2=INL2+ADD	ROFF	1881
	END=END+ADD	ROFF	1882
3	PUT IN CHANGE WORD	ROFF	1883
	IF (CORRECT(2,I).EQ.0) GO TO 15	ROFF	1884
13	COR2=COR+ADD	ROFF	1885
	DO 14 K=J,COR2	ROFF	1886
14	IN(K)=CORRECT(13+K-J,I)	ROFF	1887
15	CONTINUE	ROFF	1888
	RETURN	ROFF	1889
	END	ROFF	1890

AFWL-TR-72-139

SUBROUTINE FILL

SUBROUTINE FILL			
2	IMPLICIT INTEGER(A-Z)	ROFF	1891
	INTEGER BLANK, BUFFL, B1, CC, OLENG, OUT, OVLINE, PAGEL, PAGENO, PCC, PSW, RN	ROFF	1892
	1UNSH, SB, START, ULINE	ROFF	1893
	INTEGER COLUMN	ROFF	1894
	COMMON /INBUF/ IN(99), ULINE(99), PRU, INLENG, INL1	ROFF	1895
	COMMON /OPARY/ CC, PCC, INDENT, PAGENO, LINECT, PAGFL, PHONSH, RNUMSH	ROFF	1896
	LOGICAL PHONSH	ROFF	1897
	COMMON /OUTBJ/ OUT(130), OVLINE(130), BUFFL, OVERSH, NWORD, OLENG, PSW,	ROFF	1898
	LENMAX	ROFF	1899
	COMMON /SWTCH/ ADSW, FILSW	ROFF	1900
	LOGICAL PRU, OVERSH, ADSW	ROFF	1901
	LOGICAL FILLSH, SPELSH	ROFF	1902
	COMMON /SR/ SPELSH	ROFF	1903
	COMMON /SR/ CJLUMN, INL2	ROFF	1904
	JATA BLANK/6/	ROFF	1905
	INL2=INL1	ROFF	1906
	IF (.NOT. SPELSH) GO TO 1	ROFF	1907
	CALL SPELL (IN, INL1)	ROFF	1908
2	FIND FIRST NON-BLANK IN THE LINE	ROFF	1909
1	DO 2 I=1, INL2	ROFF	1910
	IF (IN(I).NE.BLANK) GO TO 3	ROFF	1911
2	CONTINUE	ROFF	1912
3	GET HERE IF INPUT IS A BLANK LINE	ROFF	1913
	RETURN	ROFF	1914
4	FOUND NON-BLANK. LOOK FOR BLANK	ROFF	1915
3	START=I	ROFF	1916
	DO 4 I=START, INL2	ROFF	1917
	IF (IN(I).EQ.BLANK) GO TO 5	ROFF	1918
4	CONTINUE	ROFF	1919
5	GET HERE FOR BLANK	ROFF	1920
6	LWORD=I-START	ROFF	1921
7	TOO BIG FOR BUFFER	ROFF	1922
	BUFFL IS LAST BLANK IN OUTPUT BUFFER	ROFF	1923
8	IF (LWORD+BUFFL.GT.OLENG) GO TO 12	ROFF	1924
9	NO, SO PUT WORD INTO BUFFER	ROFF	1925
7	NWORD=NWORD+1	ROFF	1926
	SB=START-BUFFL-1	ROFF	1927
	B1=BUFFL+1	ROFF	1928
	BUFFL=BUFFL+LWORD+1	ROFF	1929
	DO 8 J=B1, BUFFL	ROFF	1930
	OUT(J)=IN(SB+J)	ROFF	1931
9	CONTINUE	ROFF	1932
10	PUT IN OVERSTRICKE LINE IF NEEDED	ROFF	1933
	IF (.NOT. PRU) GO TO 10	ROFF	1934
11	SET OVERSTRICKE SW TO REMEMBER FOR OUTPUT	ROFF	1935
	OVERSH=.TRUE.	ROFF	1936
	DO 9 J=B1, BUFFL	ROFF	1937
	OVLINE(J)=ULINE(SB+J)	ROFF	1938
		ROFF	1939
	LOOK FOR NEXT NON-BLANK. CAREFUL ABOUT FALLING OFF END	ROFF	1940
10	START=I	ROFF	1941
	DO 11 I=START, INL1	ROFF	1942
	IF (IN(I).NE.BLANK) GO TO 3	ROFF	1943
11	CONTINUE	ROFF	1944
		ROFF	1945

AFWL-TR-72-139

SUBROUTINE FILL

3	END OF LINE READING BLANKS. IF GET HERE. QUIT	ROFF	1946
	IF (PRU) GO TO 14	ROFF	1947
	RETURN	ROFF	1948
3		ROFF	1949
3	COME HERE TO CHECK FOR RJUST.	ROFF	1950
12	CONTINUE	ROFF	1951
	IF (IN(I-1).E2.1.AND.LWORD-1+BUFFL.E2.OLENG) GO TO 16	ROFF	1952
	IF (OUT(BUFL-1).NE.1) GO TO 13	ROFF	1953
3	KNUCK OUT EXTRA SPACE AFTER PERIOD IF AT LINE END	ROFF	1954
	BUFL=BUFL-1	ROFF	1955
	OUT(BUFL)=BLANK	ROFF	1956
13	IF (AUSW) CALL ADJUST	ROFF	1957
3	FLUSH THE OUTPUT BUFFER	ROFF	1958
	CALL FLUSH	ROFF	1959
3	AND TRY THE LAST WORD AGAIN	ROFF	1960
	IF (LWORD+BUFL.E2.OLENG) GO TO 7	ROFF	1961
3	WORD IS TOO BIG FOR LINE -- BREAK IT UP	ROFF	1962
	LWORD=OLENG-BUFL	ROFF	1963
	I=START+OLENG	ROFF	1964
	GO TO 6	ROFF	1965
3	BLANK OUT UNDERLINE	ROFF	1966
14	PRU=.FALSE.	ROFF	1967
	DO 15 I=1,INLENG	ROFF	1968
15	ULINE(I)=BLANK	ROFF	1969
	RETURN	ROFF	1970
3	PERIOD COMES RIGHT TO LINE END - KILL EXTRA SPACE	ROFF	1971
16	LWORD=LWORD-1	ROFF	1972
	IN(I-1)=BLANK	ROFF	1973
	GO TO 7	ROFF	1974
	END	ROFF	1975

AFWL-TR-72-139

	SUBROUTINE CRRECT		
	SUBROUTINE CRRECT		
2	IMPLICIT INTEGER (A-Z)	ROFF	1976
	COMMON /INBUF/ IN(99),ULINE(99),PRU,INLENG,INL1	ROFF	1977
	INTEGER BLANK,CHAR,ULINE	ROFF	1978
	LOGICAL PRU	ROFF	1979
	DATA LNOT,BLANK/95,24/	ROFF	1980
	JJ=1	ROFF	1981
	DO 2 I=1,80	ROFF	1982
	CHAR=IN(I)	ROFF	1983
	IF (CHAR.EQ.LNOT) GO TO 1	ROFF	1984
	IN(JJ)=CHAR	ROFF	1985
	JJ=JJ+1	ROFF	1986
	GO TO 2	ROFF	1987
1	IF (JJ.EQ.1) GO TO 2	ROFF	1988
	JJ=JJ-1	ROFF	1989
2	CONTINUE	ROFF	1990
	DO 3 I=JJ,80	ROFF	1991
3	IN(I)=BLANK	ROFF	1992
	RETURN	ROFF	1993
	END	ROFF	1994
		ROFF	1995

AFWL-TR-72-139

FUNCTION IRV

INTEGER FUNCTION IRV(DUMMY)

INTEGER DUMMY

X=RANF(J)

X IS UNIFORM ON 0,1 SO IRV IS EITHER 0 OR 1

IRV=X+0.5

RETURN

END

ROFF	1996
ROFF	1997
ROFF	1998
ROFF	1999
ROFF	2000
ROFF	2001
ROFF	2002

SUBROUTINE EQROFF				
SUBROUTINE EQROFF			ROFF	2003
IMPLICIT INTEGER (A-Z)			ROFF	2004
INTEGER BOTH,3PLACE,DIFF,EX,EXC,FINAL,FO,OB,OVLIN,PLACE,PL5,RP,RP			ROFF	2005
1PLACE,TOP,ULINE,UUT			ROFF	2006
INTEGER BUFF,ZMOVE			ROFF	2007
LOGICAL PSW			ROFF	2008
COMMON /INBUF/ IN(99),ULINE(99),PRU,INLENG,INL1			ROFF	2009
COMMON /EQBU/ FINAL(200,4),LMIN,LMAX,EQSW			ROFF	2010
COMMON /OUTBUF/ UUT(130),OVLIN(130),BUFFL,OVERSW,NWORD,LL,PSW,LEN			ROFF	2011
1M1X			ROFF	2012
LOGICAL OVERSW,PRU			ROFF	2013
LOGICAL ADWS,FILLSW			ROFF	2014
COMMON /SWITCH/ ADWS,FILLSW			ROFF	2015
LOGICAL EQSW			ROFF	2016
DIMENSION LEN,TH(4)			ROFF	2017
INTEGER POS,J-IN,OP5,FRA2,Q2,Q4,Z,PP,OEND,P5,Y,UP			ROFF	2018
INTEGER FRACO(2,50,2),OUT(10,500)			ROFF	2019
LOGICAL CRASH,SUPSW,SUBSW,RJSH,CESW,JIDEN,OVLIN,ATSW,OVCK			ROFF	2020
INTEGER AMPER,ATSIGN,BLANK,CENT,CFLEX,COLON,DASH,DOLLAR,EXCLAM,GRA			ROFF	2021
1VE,PERIOD,QM,QUOTE2,SHARP,LNOT,MP1211,OB2,PERCEN,USCORE			ROFF	2022
DATA MC/0/,FRAC/0/,BOTH/0/,LINE/3/			ROFF	2023
DATA FRACO/200*0/,OUT/5000*0/			ROFF	2024
DATA CRASH/.FALSE./			ROFF	2025
DATA SUPSW/.FALSE./,SUBSW/.FALSE./,RJSH/.FALSE./,CESW/.FALSE./,DIO			ROFF	2026
1GEN/.FALSE./,OVLIN/.FALSE./,ATSW/.FALSE./,OVCK/.FALSE./			ROFF	2027
DATA AMPER/80/,ATSIGN/124/,BLANK/64/,CENT/74/,CFLEX/190/,COLON/122			ROFF	2028
1/,DASH/96/,DOLLAR/91/,EXCLAM/90/,GRAVE/121/,LNOT/95/,MP1211/106/,0			ROFF	2029
282/224/,PERCEN/108/,PERIOD/75/,QM/111/,QUOTE2/127/,SHARP/123/,USCO			ROFF	2030
3RE/109/			ROFF	2031
IF (FILLSW) CALL FLUSH			ROFF	2032
CALL WRBLNK (2)			ROFF	2033
EQSW=.TRUE.			ROFF	2034
DO 3 J=1,200			ROFF	2035
DO 3 I=1,4			ROFF	2036
3 FINAL(J,I)=BLANK			ROFF	2037
DO 4 K=1,10			ROFF	2038
DO 4 I=5,500,5			ROFF	2039
4 OUT(K,I)=BLANK			ROFF	2040
DO 5 I=1,80			ROFF	2041
5 IN(I)=BLANK			ROFF	2042
5 CALL PRE (IN,1,INLENG,IEO)			ROFF	2043
CALL CORRECT			ROFF	2044
IF (IN(1).EQ.PERIOD) GO TO 43			ROFF	2045
			ROFF	2046
			ROFF	2047
DO 48 II=1,80			ROFF	2048
INIIII=IN(II)			ROFF	2049
IF (INIIII.EQ.BLANK) GO TO 43			ROFF	2050
IF (ATSW) GO TO 7			ROFF	2051
IF (INIIII.EQ.QM) GO TO 12			ROFF	2052
IF (INIIII.EQ.QUOTE2) GO TO 13			ROFF	2053
IF (INIIII.EQ.AMPER) GO TO 26			ROFF	2054
IF (INIIII.EQ.MP1211) GO TO 44			ROFF	2055
IF (INIIII.EQ.SHARP) GO TO 18			ROFF	2056
IF (INIIII.EQ.GRAVE) GO TO 38			ROFF	2057

AFWL-TR-72-139

SUBROUTINE EQROFF

	IF (INIIII.EQ.082) GO TO 14	ROFF	2058
	IF (INIIII.EQ.CENT) GO TO 9	ROFF	2059
	IF (INIIII.EQ.USCORE) GO TO 10	ROFF	2060
	IF (INIIII.EQ.ATSIGN) GO TO 11	ROFF	2061
	IF (INIIII.EQ.DOLLAR) GO TO 9	ROFF	2062
	IF (INIIII.EQ.CFLEX) GO TO 9	ROFF	2063
	MC=MIN0(MC+1,99)	ROFF	2064
	POS=5*MC	ROFF	2065
	IF (INIIII.EQ.PERIOD.OR.INIIII.EQ.EXCLAM.OR.INIIII.EQ.COLON) OUT(L	ROFF	2066
	LINE,POS+4)=DOLLAR	ROFF	2067
	GO TO 8	ROFF	2068
7	POS=5*MC+2	ROFF	2069
	ATSW=.FALSE.	ROFF	2070
9	OUT(LINE,POS)=INIIII	ROFF	2071
	GO TO 48	ROFF	2072
9	POS=5*MC+4	ROFF	2073
	GO TO 8	ROFF	2074
10	POS=5*MC+3	ROFF	2075
	GO TO 8	ROFF	2076
11	POS=5*MC+1	ROFF	2077
	ATSW=.TRUE.	ROFF	2078
	GO TO 8	ROFF	2079
:		ROFF	2080
:		ROFF	2081
12	IF (FRAC.NE.0.OR.BOTH.EQ.1) GO TO 48	ROFF	2082
	SUPSW=.NOT.SJ3SW	ROFF	2083
	IF (SUPSW) LINE=LINE-1	ROFF	2084
	IF (.NOT.SUPSW) LINE=LINE+1	ROFF	2085
	GO TO 48	ROFF	2086
13	IF (FRAC.NE.0.OR.BOTH.EQ.2) GO TO 48	ROFF	2087
	SUBSW=.NOT.SJ3SW	ROFF	2088
	IF (SUBSW) LINE=LINE+1	ROFF	2089
	IF (.NOT.SUBSW) LINE=LINE-1	ROFF	2090
	GO TO 48	ROFF	2091
14	IF (FRAC.NE.0) GO TO 48	ROFF	2092
	OVLSW=.FALSE.	ROFF	2093
	BOTH=BOTH+1	ROFF	2094
	IF (BOTH-2) 15,16,17	ROFF	2095
15	PLACE=MC	ROFF	2096
	IF (SUPSW.OR.SUBSW) GO TO 48	ROFF	2097
	LINE=LINE-1	ROFF	2098
	SUPSW=.TRUE.	ROFF	2099
	GO TO 48	ROFF	2100
16	TOP=MC	ROFF	2101
	MC=PLACE	ROFF	2102
	IF (SUBSW.OR..NOT.SUPSW) GO TO 48	ROFF	2103
	LINE=LINE+2	ROFF	2104
	SUPSW=.FALSE.	ROFF	2105
	SUBSW=.TRUE.	ROFF	2106
	GO TO 48	ROFF	2107
17	SUBSW=.FALSE.	ROFF	2108
	SUPSW=.FALSE.	ROFF	2109
	BOTH=0	ROFF	2110
	MC=MIN0(MAX0(MC,TOP),99)	ROFF	2111
	LINE=3	ROFF	2112



SUBROUTINE EQROFF

	IF (RJSH.OR.OJSH) LINE=9	ROFF	2113
	GO TO 48	ROFF	2114
18	OVLSH=.NOT.OVLSH	ROFF	2115
	IF (.NOT.OVLSH) GO TO 19	ROFF	2116
	OP5=5*(MC+1)	ROFF	2117
	GO TO 48	ROFF	2118
19	IF (FRAC.NE.J) GO TO 25	ROFF	2119
	OLIN=LINE-1	ROFF	2120
	OP5=MIN0(OP5,495)	ROFF	2121
	OUT(OLIN,OP5-2)=USCORE	ROFF	2122
	MC5=MAX0(5*MC,OP5)	ROFF	2123
	IF (BOTH.EQ.2) GO TO 22	ROFF	2124
	DO 20 I=OP5,MC5,5	ROFF	2125
20	OUT(OLIN,I)=PERCEN	ROFF	2126
21	OUT(OLIN,MC5+1)=USCORE	ROFF	2127
	GO TO 48	ROFF	2128
22	DO 23 I=OP5,MC5,5	ROFF	2129
	IF (OUT(OLIN,I).EQ.BLANK) OUT(OLIN,I)=PERCEN	ROFF	2130
23	CONTINUE	ROFF	2131
	IF (OUT(OLIN,MC5+1).EQ.ATSIGN) GO TO 24	ROFF	2132
	GO TO 21	ROFF	2133
24	OUT(OLIN,MC5+1)=082	ROFF	2134
	GO TO 48	ROFF	2135
25	JF=JF+1	ROFF	2136
	FRACO(FRAC,JF,1)=OP5	ROFF	2137
	FRACO(FRAC,JF,2)=MAX0(5*MC,OP5)	ROFF	2138
	GO TO 48	ROFF	2139
26	FRAC=FRAC+1	ROFF	2140
	IF (FRAC-2) 27,28,29	ROFF	2141
27	BOTH=0	ROFF	2142
	SUPSH=.FALSE.	ROFF	2143
	SUBSH=.FALSE.	ROFF	2144
	OVLSH=.FALSE.	ROFF	2145
	PLACE=MC	ROFF	2146
	MC=0	ROFF	2147
	LINE=5	ROFF	2148
	JF=0	ROFF	2149
	GO TO 48	ROFF	2150
28	TOP=MC	ROFF	2151
	MC=0	ROFF	2152
	LINE=6	ROFF	2153
	JF=0	ROFF	2154
	GO TO 48	ROFF	2155
29	FRAC=0	ROFF	2156
	ZMOVE=0	ROFF	2157
	LINE=3	ROFF	2158
	IF (RJSH.OR.OJSH) LINE=9	ROFF	2159
	Q2=5*PLACE	ROFF	2160
	Q4=Q2	ROFF	2161
	DIFF=MC-TOP	ROFF	2162
	EX=((IABS(DIFF)+1)/2)*5	ROFF	2163
	IF (DIFF) 30,32,31	ROFF	2164
30	Q4=Q4+EX	ROFF	2165
	ZMOVE=2	ROFF	2166
	MC=TOP	ROFF	2167

## SUBROUTINE EQROFF

	GO TO 32	ROFF	2168
31	Q2=Q2+EX	ROFF	2169
	ZMOVE=1	ROFF	2170
32	MD=MIN0(5*MC+4,5J0-MAXJ(Q2,Q4))	ROFF	2171
	DO 33 J=3,MD	ROFF	2172
	OUT(LINE-1,J+Q2)=OUT(5,J)	ROFF	2173
	OUT(LINE+1,J+Q4)=OUT(6,J)	ROFF	2174
	DO 33 Z=5,6	ROFF	2175
	OUT(Z,J)=0	ROFF	2176
	IF (MOD(J,2).EQ.1) OUT(Z,J)=BLANK	ROFF	2177
33	CONTINUE	ROFF	2178
	MC=MIN0(MC+PLACE,99)	ROFF	2179
	MCJ=2*MC	ROFF	2180
	PLJ=5*PLACE	ROFF	2181
	PP=MIN0(PLJ+2,495)	ROFF	2182
	MCJ=MAXJ(MCJ,PP)	ROFF	2183
	DO 34 I=PP,MCJ,5	ROFF	2184
34	OUT(LINE,1)=JASH	ROFF	2185
	DO 37 Z=1,2	ROFF	2186
	ULIN=LINE+2*Z-4	ROFF	2187
	DO 36 JE=1,5	ROFF	2188
	FO=FRACO(Z,JE,1)	ROFF	2189
	IF (FO.EQ.1) GO TO 37	ROFF	2190
	MOVE=PLJ	ROFF	2191
	IF (Z.EQ.1) MOVE=MOVE+EX	ROFF	2192
	OPJ=MIN0(FO+MOVE,495)	ROFF	2193
	OEND=MIN0(FRACO(Z,JE,2)+MOVE,495)	ROFF	2194
	FRACO(Z,JE,1)=0	ROFF	2195
	FRACO(Z,JE,2)=0	ROFF	2196
	OUT(ULIN,OPJ-2)=JSCORE	ROFF	2197
	OUT(OLIN,OEND+1)=USCORE	ROFF	2198
	IF (Z.EQ.2) GO TO 36	ROFF	2199
	DO 35 Y=OPJ,OEND,5	ROFF	2200
35	OUT(OLIN,Y)=2-RCEN	ROFF	2201
36	CONTINUE	ROFF	2202
37	CONTINUE	ROFF	2203
	GO TO 43	ROFF	2204
38	IF (GESH.OR.FRAC.NE.0) GO TO 47	ROFF	2205
	OVLSW=.FALSE.	ROFF	2206
	RJSH=.NOT.RJSH	ROFF	2207
	IF (.NOT.RJSH) GO TO 39	ROFF	2208
	RPLAC=AC	ROFF	2209
	MC=0	ROFF	2210
	LINE=9	ROFF	2211
	GO TO 48	ROFF	2212
39	PLACE=LL-2-MC	ROFF	2213
	RP=MIN0(RPLAC+2,99)	ROFF	2214
	IF (PLACE.GE.RP) GO TO 42	ROFF	2215
	IF (QIDGEN.ANJ.CPLACE.NE.0) GO TO 40	ROFF	2216
	PLACE=RP	ROFF	2217
	GO TO 42	ROFF	2218
40	EXC=MIN0(RP-PLACE,CPLACE)	ROFF	2219
	NUCP=CPLACE-EXC	ROFF	2220
	NJ=5*NUCP	ROFF	2221
	DO 41 I=1,4	ROFF	2222

Reproduced from  
best available copy.

AFWL-TR-72-139

SUBROUTINE EQROFF

	DO 41 J=3,MD	ROFF	2223
	JS=J+P5	ROFF	2224
	OUT(I,J+NS)=OJT(I,JS)	ROFF	2225
	OUT(I,JS)=	ROFF	2226
+1	IF (MOD(JS,5).EQ.0) OUT(I,JS)=BLANK	ROFF	2227
	CONTINUE	ROFF	2228
	PLAGE=MAX0(PLAGE,RP-EXC)	ROFF	2229
+2	MD=5*MC+4	ROFF	2230
	MC=MIN0(MC+PLAGE,99)	ROFF	2231
	P5=5*PLAGE	ROFF	2232
	MD=MIN0(MD,50)-P5)	ROFF	2233
	DO 43 I=1,4	ROFF	2234
	I6=I+6	ROFF	2235
	DO 43 J=3,MD	ROFF	2236
	OUT(I,J+P5)=OJT(I6,J)	ROFF	2237
	OUT(I6,J)=J	ROFF	2238
+3	IF (MOD(J,5).EQ.0) OUT(I6,J)=BLANK	ROFF	2239
	CONTINUE	ROFF	2240
	LINE=3	ROFF	2241
	GO TO 48	ROFF	2242
+4	IF (RJSH.OR.FRAG.NE.0) GO TO 47	ROFF	2243
	OVLSH=.FALSE.	ROFF	2244
	CESH=.NOT.CESH	ROFF	2245
	IF (.NOT.CESH) GO TO 45	ROFF	2246
	LPLACE=MC	ROFF	2247
	MC=0	ROFF	2248
	LINE=9	ROFF	2249
	GO TO 48	ROFF	2250
+5	CPLACE=(LL-MC)/2-2	ROFF	2251
	CPLACE=MAX0(CPLACE,LPLACE+2)	ROFF	2252
	MD=5*MC+4	ROFF	2253
	MC=MIN0(MC+CPLACE,99)	ROFF	2254
	P5=5*CPLACE	ROFF	2255
	MD=MIN0(MD,50)-P5)	ROFF	2256
	DO 46 I=1,4	ROFF	2257
	I6=I+6	ROFF	2258
	DO 46 J=3,MD	ROFF	2259
	OUT(I,J+P5)=OJT(I6,J)	ROFF	2260
	OUT(I6,J)=J	ROFF	2261
	IF (MOD(J,5).EQ.0) OUT(I6,J)=BLANK	ROFF	2262
+6	CONTINUE	ROFF	2263
	DIOGEN=.TRUE.	ROFF	2264
	LINE=3	ROFF	2265
	GO TO 48	ROFF	2266
+7	CRASH=.TRUE.	ROFF	2267
	MC=MAX0(MC,1)	ROFF	2268
	OUT(1,5*MC)=STAR	ROFF	2269
+8	CONTINUE	ROFF	2270
	GO TO 6	ROFF	2271
		ROFF	2272
		ROFF	2273
+9	CONTINUE	ROFF	2274
	DO 53 K=1,4	ROFF	2275
	FINAL(1,K)=DOLLAR	ROFF	2276
	L=4	ROFF	2277

AFNL-TR-72-139

SUBROUTINE EQROFF

```

LEN=5*MC+4
DO 50 J=3,LEN
OB=OUT(K,J)
IF (OB.EQ.0) GO TO 50
FINAL(L,K)=OB
L=L+1
50 CONTINUE
L1=L-1
LENGTH(K)=L1
DO 51 J=1,L1
IF (FINAL(L-J,K).NE.BLANK) GO TO 52
51 LENGTH(K)=LENGTH(K)-1
52 IF (LENGTH(K).EQ.1) GO TO 53
INLENG=MAX0(INLENG,LENGTH(K))
IF (K.GT.LMAX) LMAX=K
IF (K.LT.LMIN) LMIN=K
53 CONTINUE
UP=4
IF (CRASH) UP=10
DO 54 K=1,UP
DO 54 J=1,500
OUT(K,J)=0
IF (MOD(J,5).EQ.0) OUT(K,J)=BLANK
54 CONTINUE
FRAC=0
BOTH=0
SUPSW=.FALSE.
SUBSW=.FALSE.
OVLSW=.FALSE.
RJSW=.FALSE.
CESW=.FALSE.
ATSW=.FALSE.
JIDGEN=.FALSE.
CRASH=.FALSE.
MC=0
LINE=3
RETURN
END

```

```

ROFF      2278
ROFF      2279
ROFF      2280
ROFF      2281
ROFF      2282
ROFF      2283
ROFF      2284
ROFF      2285
ROFF      2286
ROFF      2287
ROFF      2288
ROFF      2289
ROFF      2290
ROFF      2291
ROFF      2292
ROFF      2293
ROFF      2294
ROFF      2295
ROFF      2296
ROFF      2297
ROFF      2298
ROFF      2299
ROFF      2300
ROFF      2301
ROFF      2302
ROFF      2303
ROFF      2304
ROFF      2305
ROFF      2306
ROFF      2307
ROFF      2308
ROFF      2309
ROFF      2310
ROFF      2311
ROFF      2312
ROFF      2313
ROFF      2314
ROFF      2315

```

# AFWL-TR-72-139

## SUBROUTINE MICRO

SUBROUTINE MICRO (ICC, ID, IJ)	ROFF	2316
LOGICAL SECONJ	ROFF	2317
DIMENSION IDATA(256), ID(132), IT(132)	ROFF	2318
DIMENSION IITBL(256)	ROFF	2319
INTEGER PLUS, ONE, ZERO	ROFF	2320
COMMON /FRAME/ IFRAME	ROFF	2321
COMMON /TAPE/ ITAPE	ROFF	2322
DATA PLUS, ONE, ZERO/1H+, 141, 140/	ROFF	2323
DATA IREG//	ROFF	2324
DATA IFRAME//	ROFF	2325
DATA SECONJ/.FALSE./	ROFF	2326
DATA IITBL(1)/1338/	ROFF	2327
DATA IITBL(2)/0558/	ROFF	2328
DATA IITBL(3)/0558/	ROFF	2329
DATA IITBL(4)/0558/	ROFF	2330
DATA IITBL(5)/0558/	ROFF	2331
DATA IITBL(6)/0558/	ROFF	2332
DATA IITBL(7)/0558/	ROFF	2333
DATA IITBL(8)/0558/	ROFF	2334
DATA IITBL(9)/0718/	ROFF	2335
DATA IITBL(10)/0558/	ROFF	2336
DATA IITBL(11)/1638/	ROFF	2337
DATA IITBL(12)/0558/	ROFF	2338
DATA IITBL(13)/0558/	ROFF	2339
DATA IITBL(14)/558/	ROFF	2340
DATA IITBL(15)/1338/	ROFF	2341
DATA IITBL(16)/0558/	ROFF	2342
DATA IITBL(17)/0558/	ROFF	2343
DATA IITBL(18)/0558/	ROFF	2344
DATA IITBL(19)/0558/	ROFF	2345
DATA IITBL(20)/1728/	ROFF	2346
DATA IITBL(21)/0558/	ROFF	2347
DATA IITBL(22)/0558/	ROFF	2348
DATA IITBL(23)/0558/	ROFF	2349
DATA IITBL(24)/0558/	ROFF	2350
DATA IITBL(25)/0558/	ROFF	2351
DATA IITBL(26)/0558/	ROFF	2352
DATA IITBL(27)/0558/	ROFF	2353
DATA IITBL(28)/0558/	ROFF	2354
DATA IITBL(29)/0558/	ROFF	2355
DATA IITBL(30)/0558/	ROFF	2356
DATA IITBL(31)/0558/	ROFF	2357
DATA IITBL(32)/0558/	ROFF	2358
DATA IITBL(33)/0558/	ROFF	2359
DATA IITBL(34)/1668/	ROFF	2360
DATA IITBL(35)/0658/	ROFF	2361
DATA IITBL(36)/0558/	ROFF	2362
DATA IITBL(37)/0558/	ROFF	2363
DATA IITBL(38)/1658/	ROFF	2364
DATA IITBL(39)/0558/	ROFF	2365
DATA IITBL(40)/0558/	ROFF	2366
DATA IITBL(41)/1768/	ROFF	2367
DATA IITBL(42)/0558/	ROFF	2368
DATA IITBL(43)/0558/	ROFF	2369
DATA IITBL(44)/0558/	ROFF	2370

## SUBROUTINE MICRO

DATA IITBL(45)/0558/  
 DATA IITBL(46)/0558/  
 DATA IITBL(47)/0558/  
 DATA IITBL(48)/0558/  
 DATA IITBL(49)/0558/  
 DATA IITBL(50)/0558/  
 DATA IITBL(51)/0558/  
 DATA IITBL(52)/0558/  
 DATA IITBL(53)/0558/  
 DATA IITBL(54)/0558/  
 DATA IITBL(55)/0558/  
 DATA IITBL(56)/0558/  
 DATA IITBL(57)/0558/  
 DATA IITBL(58)/0558/  
 DATA IITBL(59)/0558/  
 DATA IITBL(60)/0558/  
 DATA IITBL(61)/0558/  
 DATA IITBL(62)/0558/  
 DATA IITBL(63)/0558/  
 DATA IITBL(64)/0558/  
 DATA IITBL(65)/1448/  
 DATA IITBL(66)/1608/  
 DATA IITBL(67)/1378/  
 DATA IITBL(68)/1428/  
 DATA IITBL(69)/0558/  
 DATA IITBL(70)/0558/  
 DATA IITBL(71)/0558/  
 DATA IITBL(72)/0558/  
 DATA IITBL(73)/0558/  
 DATA IITBL(74)/0558/  
 DATA IITBL(75)/0578/  
 DATA IITBL(76)/0748/  
 DATA IITBL(77)/0518/  
 DATA IITBL(78)/0438/  
 DATA IITBL(79)/0558/  
 DATA IITBL(80)/0558/  
 DATA IITBL(81)/0558/  
 DATA IITBL(82)/0558/  
 DATA IITBL(83)/1438/  
 DATA IITBL(84)/1628/  
 DATA IITBL(85)/0558/  
 DATA IITBL(86)/0558/  
 DATA IITBL(87)/1408/  
 DATA IITBL(88)/1413/  
 DATA IITBL(89)/1458/  
 DATA IITBL(90)/0678/  
 DATA IITBL(91)/0728/  
 DATA IITBL(92)/0478/  
 DATA IITBL(93)/0528/  
 DATA IITBL(94)/0778/  
 DATA IITBL(95)/0768/  
 DATA IITBL(96)/0468/  
 DATA IITBL(97)/0508/  
 DATA IITBL(98)/1618/  
 DATA IITBL(99)/0558/

ROFF 2371  
 ROFF 2372  
 ROFF 2373  
 ROFF 2374  
 ROFF 2375  
 ROFF 2376  
 ROFF 2377  
 ROFF 2378  
 ROFF 2379  
 ROFF 2380  
 ROFF 2381  
 ROFF 2382  
 ROFF 2383  
 ROFF 2384  
 ROFF 2385  
 ROFF 2386  
 ROFF 2387  
 ROFF 2388  
 ROFF 2389  
 ROFF 2390  
 ROFF 2391  
 ROFF 2392  
 ROFF 2393  
 ROFF 2394  
 ROFF 2395  
 ROFF 2396  
 ROFF 2397  
 ROFF 2398  
 ROFF 2399  
 ROFF 2400  
 ROFF 2401  
 ROFF 2402  
 ROFF 2403  
 ROFF 2404  
 ROFF 2405  
 ROFF 2406  
 ROFF 2407  
 ROFF 2408  
 ROFF 2409  
 ROFF 2410  
 ROFF 2411  
 ROFF 2412  
 ROFF 2413  
 ROFF 2414  
 ROFF 2415  
 ROFF 2416  
 ROFF 2417  
 ROFF 2418  
 ROFF 2419  
 ROFF 2420  
 ROFF 2421  
 ROFF 2422  
 ROFF 2423  
 ROFF 2424  
 ROFF 2425

SUBROUTINE MICRO

DATA IITBL(100)/0558/  
 DATA IITBL(101)/0558/  
 DATA IITBL(102)/1548/  
 DATA IITBL(103)/0558/  
 DATA IITBL(104)/0558/  
 DATA IITBL(105)/0558/  
 DATA IITBL(106)/0558/  
 DATA IITBL(107)/0558/  
 DATA IITBL(108)/0628/  
 DATA IITBL(109)/0009/  
 DATA IITBL(110)/0738/  
 DATA IITBL(111)/1778/  
 DATA IITBL(112)/1758/  
 DATA IITBL(113)/1738/  
 DATA IITBL(114)/0558/  
 DATA IITBL(115)/0558/  
 DATA IITBL(116)/0558/  
 DATA IITBL(117)/1468/  
 DATA IITBL(118)/0558/  
 DATA IITBL(119)/0558/  
 DATA IITBL(120)/0558/  
 DATA IITBL(121)/0558/  
 DATA IITBL(122)/0608/  
 DATA IITBL(123)/0558/  
 DATA IITBL(124)/0558/  
 DATA IITBL(125)/1758/  
 DATA IITBL(126)/0548/  
 DATA IITBL(127)/1528/  
 DATA IITBL(128)/0558/  
 DATA IITBL(129)/1018/  
 DATA IITBL(130)/1028/  
 DATA IITBL(131)/1038/  
 DATA IITBL(132)/1048/  
 DATA IITBL(133)/1058/  
 DATA IITBL(134)/1068/  
 DATA IITBL(135)/1078/  
 DATA IITBL(136)/1108/  
 DATA IITBL(137)/1118/  
 DATA IITBL(138)/0558/  
 DATA IITBL(139)/0558/  
 DATA IITBL(140)/0558/  
 DATA IITBL(141)/0558/  
 DATA IITBL(142)/0558/  
 DATA IITBL(143)/0558/  
 DATA IITBL(144)/0558/  
 DATA IITBL(145)/1128/  
 DATA IITBL(146)/1138/  
 DATA IITBL(147)/1148/  
 DATA IITBL(148)/1158/  
 DATA IITBL(149)/1168/  
 DATA IITBL(150)/1178/  
 DATA IITBL(151)/1208/  
 DATA IITBL(152)/1218/  
 DATA IITBL(153)/1228/  
 DATA IITBL(154)/0558/

ROFF 2426  
 ROFF 2427  
 ROFF 2428  
 ROFF 2429  
 ROFF 2430  
 ROFF 2431  
 ROFF 2432  
 ROFF 2433  
 ROFF 2434  
 ROFF 2435  
 ROFF 2436  
 ROFF 2437  
 ROFF 2438  
 ROFF 2439  
 ROFF 2440  
 ROFF 2441  
 ROFF 2442  
 ROFF 2443  
 ROFF 2444  
 ROFF 2445  
 ROFF 2446  
 ROFF 2447  
 ROFF 2448  
 ROFF 2449  
 ROFF 2450  
 ROFF 2451  
 ROFF 2452  
 ROFF 2453  
 ROFF 2454  
 ROFF 2455  
 ROFF 2456  
 ROFF 2457  
 ROFF 2458  
 ROFF 2459  
 ROFF 2460  
 ROFF 2461  
 ROFF 2462  
 ROFF 2463  
 ROFF 2464  
 ROFF 2465  
 ROFF 2466  
 ROFF 2467  
 ROFF 2468  
 ROFF 2469  
 ROFF 2470  
 ROFF 2471  
 ROFF 2472  
 ROFF 2473  
 ROFF 2474  
 ROFF 2475  
 ROFF 2476  
 ROFF 2477  
 ROFF 2478  
 ROFF 2479  
 ROFF 2480

Reproduced from  
 best available copy.

SUBROUTINE MICRO

DATA IITBL(155)/0558/	ROFF	2481
DATA IITBL(156)/0558/	ROFF	2482
DATA IITBL(157)/0558/	ROFF	2483
DATA IITBL(158)/0558/	ROFF	2484
DATA IITBL(159)/0558/	ROFF	2485
DATA IITBL(160)/0558/	ROFF	2486
DATA IITBL(161)/1648/	ROFF	2487
DATA IITBL(162)/1238/	ROFF	2488
DATA IITBL(163)/1248/	ROFF	2489
DATA IITBL(164)/1258/	ROFF	2490
DATA IITBL(165)/1268/	ROFF	2491
DATA IITBL(166)/1278/	ROFF	2492
DATA IITBL(167)/1308/	ROFF	2493
DATA IITBL(168)/1318/	ROFF	2494
DATA IITBL(169)/1328/	ROFF	2495
DATA IITBL(170)/0558/	ROFF	2496
DATA IITBL(171)/0558/	ROFF	2497
DATA IITBL(172)/0558/	ROFF	2498
DATA IITBL(173)/0558/	ROFF	2499
DATA IITBL(174)/0558/	ROFF	2500
DATA IITBL(175)/0558/	ROFF	2501
DATA IITBL(176)/0558/	ROFF	2502
DATA IITBL(177)/1348/	ROFF	2503
DATA IITBL(178)/1748/	ROFF	2504
DATA IITBL(179)/1578/	ROFF	2505
DATA IITBL(180)/0558/	ROFF	2506
DATA IITBL(181)/0558/	ROFF	2507
DATA IITBL(182)/0558/	ROFF	2508
DATA IITBL(183)/0558/	ROFF	2509
DATA IITBL(184)/0558/	ROFF	2510
DATA IITBL(185)/0558/	ROFF	2511
DATA IITBL(186)/0558/	ROFF	2512
DATA IITBL(187)/0638/	ROFF	2513
DATA IITBL(188)/0558/	ROFF	2514
DATA IITBL(189)/0648/	ROFF	2515
DATA IITBL(190)/0678/	ROFF	2516
DATA IITBL(191)/1508/	ROFF	2517
DATA IITBL(192)/0638/	ROFF	2518
DATA IITBL(193)/0018/	ROFF	2519
DATA IITBL(194)/0028/	ROFF	2520
DATA IITBL(195)/0038/	ROFF	2521
DATA IITBL(196)/0048/	ROFF	2522
DATA IITBL(197)/0058/	ROFF	2523
DATA IITBL(198)/0068/	ROFF	2524
DATA IITBL(199)/0078/	ROFF	2525
DATA IITBL(200)/0108/	ROFF	2526
DATA IITBL(201)/0118/	ROFF	2527
DATA IITBL(202)/0558/	ROFF	2528
DATA IITBL(203)/0558/	ROFF	2529
DATA IITBL(204)/0558/	ROFF	2530
DATA IITBL(205)/0558/	ROFF	2531
DATA IITBL(206)/0558/	ROFF	2532
DATA IITBL(207)/0558/	ROFF	2533
DATA IITBL(208)/0648/	ROFF	2534
DATA IITBL(209)/0128/	ROFF	2535



AFWL-TR-72-139

SUBROUTINE MICRO

DATA IITBL(210)/0138/	ROFF	2536
DATA IITBL(211)/0148/	ROFF	2537
DATA IITBL(212)/0158/	ROFF	2538
DATA IITBL(213)/0168/	ROFF	2539
DATA IITBL(214)/0178/	ROFF	2540
DATA IITBL(215)/0208/	ROFF	2541
DATA IITBL(216)/0218/	ROFF	2542
DATA IITBL(217)/0228/	ROFF	2543
DATA IITBL(218)/0558/	ROFF	2544
DATA IITBL(219)/0558/	ROFF	2545
DATA IITBL(220)/0558/	ROFF	2546
DATA IITBL(221)/0558/	ROFF	2547
DATA IITBL(222)/0558/	ROFF	2548
DATA IITBL(223)/0558/	ROFF	2549
DATA IITBL(224)/0558/	ROFF	2550
DATA IITBL(225)/0558/	ROFF	2551
DATA IITBL(226)/0238/	ROFF	2552
DATA IITBL(227)/0248/	ROFF	2553
DATA IITBL(228)/0258/	ROFF	2554
DATA IITBL(229)/0268/	ROFF	2555
DATA IITBL(230)/0278/	ROFF	2556
DATA IITBL(231)/0308/	ROFF	2557
DATA IITBL(232)/0318/	ROFF	2558
DATA IITBL(233)/0328/	ROFF	2559
DATA IITBL(234)/0558/	ROFF	2560
DATA IITBL(235)/0558/	ROFF	2561
DATA IITBL(236)/0558/	ROFF	2562
DATA IITBL(237)/0558/	ROFF	2563
DATA IITBL(238)/0558/	ROFF	2564
DATA IITBL(239)/0558/	ROFF	2565
DATA IITBL(240)/0338/	ROFF	2566
DATA IITBL(241)/0348/	ROFF	2567
DATA IITBL(242)/0358/	ROFF	2568
DATA IITBL(243)/0368/	ROFF	2569
DATA IITBL(244)/0378/	ROFF	2570
DATA IITBL(245)/0408/	ROFF	2571
DATA IITBL(246)/0418/	ROFF	2572
DATA IITBL(247)/0428/	ROFF	2573
DATA IITBL(248)/0438/	ROFF	2574
DATA IITBL(249)/04 B/	ROFF	2575
DATA IITBL(250)/0558/	ROFF	2576
DATA IITBL(251)/0558/	ROFF	2577
DATA IITBL(252)/0558/	ROFF	2578
DATA IITBL(253)/0558/	ROFF	2579
DATA IITBL(254)/0558/	ROFF	2580
DATA IITBL(255)/0558/	ROFF	2581
DATA IITBL(256)/0558/	ROFF	2582
IF (SECOND) GO TO 1	ROFF	2583
CALL SSWTCH (1,ITAPE)	ROFF	2584
CALL SSWTCH (2,IMIKE)	ROFF	2585
IF (IMIKE.EQ.1) CALL INIT (0)	ROFF	2586
SECOND=.TRUE.	ROFF	2587
CONTINUE	ROFF	2588
IF (ITAPE.EQ.1) CALL WRT9203 (ICC,IG,IJ)	ROFF	2589
IF (IMIKE.EQ.2) RETURN	ROFF	2590

1

AFWL-TR-72-139

SUBROUTINE MICRO

	IF (IJ.EQ.0) 30 TO 3	ROFF	2591
	DO 2 I=1,IJ	ROFF	2592
	J=I0(I)	ROFF	2593
	IF (J.GT.256.JR.J.LE.0) J=64	ROFF	2594
2	IT(I)=IIT8L(J)	ROFF	2595
3	IF (ICC.NE.P.US) CALL ADVAN (0)	ROFF	2596
	IF (ICC.EQ.ONE) CALL PAGER (3)	ROFF	2597
	IF (ICC.EQ.ONE) IFRAME=IFRAME+1	ROFF	2598
	IF (ICC.EQ.ZERO) CALL ADVAN (0)	ROFF	2599
	IF (IJ.EQ.0) RETURN	ROFF	2600
	CALL WRITER (IT,IJ)	ROFF	2601
	RETURN	ROFF	2602
	ENDU	ROFF	2603

## SUBROUTINE WRITER

	SUBROUTINE WRITER (IDATA,N)	ROFF	2604
	THIS SUBROUTINE TEST OUT THE MICROFILMER WITH UPPER AND LOWER CASE	ROFF	2605
	IN REPRODUCTION USE MAGNIFICATION 16	ROFF	2606
	DIMENSION IDATA(1)	ROFF	2607
	THE DATA BYT= IN IDATA IS RIGHT JUSTIFIED	ROFF	2608
	UPPER CASE A-Z OR OCTAL 01-32	ROFF	2609
	LOWER CASE A-Z OF OCTAL 101-132	ROFF	2610
	UPPER CASE ITALICS A-Z OR OCTAL 201-232	ROFF	2611
	LOWER CASE ITALICS A-Z OR OCTAL 301-332	ROFF	2612
	ALL THE ABOVE SIZE ONE	ROFF	2613
	SIZE 0 IS AS ABOVE BUT WITH BIT 4 ON	ROFF	2614
	DATA IOCAS,ITAL,0,0/	ROFF	2615
	DO 6 I=1,N	ROFF	2616
	J=IDATA(I)	ROFF	2617
	ICASE=SHIFT(J.AND.1008,-5)	ROFF	2618
	ITALIC=SHIFT(J.AND.2008,-7)	ROFF	2619
	ISIZE=SHIFT(J.AND.4008,-8)	ROFF	2620
	SET PARAMETER FOR SYMBOL AND TAB MODE	ROFF	2621
1	CONTINUE	ROFF	2622
	IF (IOCAS.EQ.ICASE.AND.ITAL.EQ.ITALIC) GO TO 2	ROFF	2623
	CALL PLOTQ (ICASE,ITALIC,0,0,3)	ROFF	2624
	IOCAS=ICASE	ROFF	2625
	ITAL=ITALIC	ROFF	2626
2	CONTINUE	ROFF	2627
	MOVE BEAM	ROFF	2628
	DA=PSUX	ROFF	2629
	IF (ISIZE.EQ.1) DA=PS1X	ROFF	2630
	A=A+DA	ROFF	2631
	J=J.AND.778	ROFF	2632
	IF (J.EQ.008) GO TO 4	ROFF	2633
	IF (J.EQ.558) GO TO 5	ROFF	2634
	J=SHIFT(J,54)	ROFF	2635
	CALL PLOTQ (B,A,0,0,2)	ROFF	2636
	PLOT CHARACTER	ROFF	2637
3	CONTINUE	ROFF	2638
	CALL PLOTQ (J,IROT,1,ISIZE,6)	ROFF	2639
	GO TO 5	ROFF	2640
4	CONTINUE	ROFF	2641
	POSITION FOR UNDERLINE AND THEN GO DO IT	ROFF	2642
	CALL PLOTQ (B+DPL1/1.50,A-DA/2.,0,0,2)	ROFF	2643
	CALL PLOTQ (B+DPL1/1.50,A+DA/2.,1,1,2)	ROFF	2644
5	CONTINUE	ROFF	2645
5	CONTINUE	ROFF	2646
	A=0.	ROFF	2647
	RETURN	ROFF	2648
	ENTRY ADVAN	ROFF	2649
	DATA IROT/1/	ROFF	2650
	FINISHED	ROFF	2651
	NOW MOVE BEAM TO NEXT LINE	ROFF	2652
	A=0.	ROFF	2653
	B=B+PL1Y	ROFF	2654
	CALL PLOTQ (A,B,0,0,2)	ROFF	2655
	RETURN	ROFF	2656
	ENTRY PAGER	ROFF	2657
	A=0.	ROFF	2658

AFWL-TR-72-139

SUBROUTINE WRITER

B=DPL1  
CALL PLOTQ (0,0,0,0,12)  
RETURN  
ENTRY INIT  
CALL PLOTQ (0,0,0,0,7)  
PSOX=1023./100.  
PSIX=1023./80.  
PL1Y=1023./53.  
JPL1=PL1Y/2.  
A=0.  
B=DPL1  
RETURN  
END

ROFF	2659
ROFF	2660
ROFF	2661
ROFF	2662
ROFF	2663
ROFF	2664
ROFF	2665
ROFF	2666
ROFF	2667
ROFF	2668
ROFF	2669
ROFF	2670
ROFF	2671

AFWL-TR-72-139

IDENT PLOTQ  
PROGRAM LENGTH

ROFF 2672

BLOCKS

PROGRAM\* LOCAL  
SHAP COMMON

ENTRY POINTS

000001 PLOTQ

EXTERNAL SYMBOLS

XRGL GET3A SYSTEM ABNORM.

CON	MACRO	A		ROFF	2673
	DATA	A		ROFF	2674
	ENDM			ROFF	2675
	EXT	XRGL, GET3A, SYSTEM, ABNORM.		ROFF	2676
	USE	/SM. P/		ROFF	2677
XMIN	BSS	1		ROFF	2678
XMAX	DATA	1		ROFF	2679
YMIN	BSS	1		ROFF	2680
YMAX	DATA	1		ROFF	2681
XMI	BSS	1		ROFF	2682
XMA	BSS	1		ROFF	2683
YMI	BSS	1		ROFF	2684
YMA	BSS	1		ROFF	2685
XSCALE	DATA	1023		ROFF	2686
YSCALE	DATA	1023		ROFF	2687
	JSE	*		ROFF	2688
	ENTRY	PLOTQ		ROFF	2689
NAME	VFD	42/0H-PLOTQ, 18/0		ROFF	2690
PLOTQ	BSS	1		ROFF	2691
*SAVE AD	AS REQUIRED BY FTM CONVENTION			ROFF	2692
	SX6	A0		ROFF	2693
	SA6	SAVA0		ROFF	2694
*GATHER JP	5 ARGUMENTS			ROFF	2695
	S87	1		ROFF	2696
	SA1	A1		ROFF	2697
	SB1	X1		ROFF	2698
	SA1	A1+87		ROFF	2699
	S82	X1		ROFF	2700
	SA1	A1+37		ROFF	2701
	S83	X1		ROFF	2702
	SA1	A1+37		ROFF	2703
	S84	X1		ROFF	2704
	SA1	A1+87		ROFF	2705
	S85	X1		ROFF	2706
	SA0	DATA+1		ROFF	2707
	SA2	B5		ROFF	2708
	SX0	B7		ROFF	2709
	SX1	X2-3		ROFF	2710
	VG	X1, SC0		ROFF	2711
	ZR	X1, SP3M	.JUMP IF SET TAB MODE PARAMETERS K = 3	ROFF	2712
	SX1	X1-2		ROFF	2713

NG	X1,PPM	.JUMP IF POINT MODE K = 4		
ZR	X1,PSYM	.JUMP IF PLOT SYMBOL ENTRY		
SX1	X1-2		K=5	ROFF 2714
NG	X1,PC	.JUMP IF PLOT CHARACTERS ENTRY		ROFF 2715
ZR	X1,PIVT	.JUMP IF RECEIVE INITIAL X,Y,IERR	K=6	ROFF 2716
SA1	X1-5		K=7	ROFF 2717
ZR	X1,PTER	.JUMP IF TERMINATE FRAME K = 12		ROFF 2718
*FALL THROUGH EXIT ILLEGAL ENTRY				ROFF 2719
PLOT1	SA1	SAVAL		ROFF 2720
	SA0	X1		ROFF 2721
	EQ	RESTORE AD		ROFF 2722
		B0,B0,PLOTQ		ROFF 2723
* PLOT POINT MODE - PLOT 1 TO 7 POINTS IN POINT MODE				ROFF 2724
PPM	SA1	B4		ROFF 2725
	SA2	B3		ROFF 2726
	SX6	30208		ROFF 2727
	ZR	X1,PPM1		ROFF 2728
	IX6	X6+X3	.ADD HIGH INTENSITY BIT	ROFF 2729
PPM1	ZR	X2,PLOT1	.EXIT IF N = 0	ROFF 2730
	SB4	X2		ROFF 2731
	-X6	608		ROFF 2732
	SA6	A0-B7	.STORE CONTROL WORD	ROFF 2733
	RJ	SCAL		ROFF 2734
	SA3	A0-B7		ROFF 2735
	LX1	308		ROFF 2736
	3X6	X1+X3		ROFF 2737
	SA6	A0-B7	.STORE FIRST SET	ROFF 2738
	EQ	B4,B0,PPM4		ROFF 2739
	RJ	SCAL		ROFF 2740
	SA3	A0-B7		ROFF 2741
	3X7	X1+X3		ROFF 2742
	SA7	A0-B7	.STORE SECOND SET	ROFF 2743
	EQ	B4,B0,PPM3		ROFF 2744
	RJ	SCAL		ROFF 2745
	LX1	448		ROFF 2746
	3X6	X1		ROFF 2747
	SA6	A0	.STORE THIRO SET	ROFF 2748
	EQ	B0,B4,PPM2		ROFF 2749
	RJ	SCAL		ROFF 2750
	SA5	A0		ROFF 2751
	-X1	148		ROFF 2752
	3X7	X5+X1		ROFF 2753
	SA7	A0	.STORE FOURTH SET	ROFF 2754
	EQ	B0,B4,PPM2		ROFF 2755
	RJ	SCAL		ROFF 2756
	LX1	608		ROFF 2757
	MX2	608		ROFF 2758
	SA3	A0		ROFF 2759
	3X7	X1*X2		ROFF 2760
	3X6	-X2*X1		ROFF 2761
	IX6	X3+X6		ROFF 2762
	SA7	A0+B7	.STORE FIFTH SET	ROFF 2763
	SA6	A3		ROFF 2764
	EQ	B0,B4,PPM3		ROFF 2765
	RJ	SCAL		ROFF 2766
	SA3	A0+B7		ROFF 2767
	LX1	308		ROFF 2768
	3X6	X1+X3		ROFF 2769
				ROFF 2770

	SA6	A3	.STORE SIXTH SET	ROFF	2771
	EQ	B0,B4,PPH5		ROFF	2772
	RJ	SCAL		ROFF	2773
	SA3	A0+B7		ROFF	2774
	3X6	X1+X3		ROFF	2775
	SA6	A3	.STORE SEVENTH SET	ROFF	2776
	EQ	B0,B0,PPH5		ROFF	2777
PPH2	SX7	B7+B7		ROFF	2778
	RJ	POUT		ROFF	2779
	EQ	B0,B0,P10F		ROFF	2780
PPH3	SX7	B7		ROFF	2781
	RJ	POUT		ROFF	2782
	EQ	B0,B0,P10F		ROFF	2783
PPH4	SX7	B0		ROFF	2784
	RJ	POUT		ROFF	2785
	EQ	B0,B0,P10F		ROFF	2786
PPH5	SX7	3		ROFF	2787
	RJ	POUT		ROFF	2788
	EQ	B0,B0,P10F		ROFF	2789
*. DO SCALING PLOTTING A=X, B=Y, I=IPEN, J=INTEN, K=0					
SCP	SA1	B4		ROFF	2790
	SA3	B3		ROFF	2791
	LX1	72B		ROFF	2792
	SB4	X3		ROFF	2793
	SX6	3210B		ROFF	2794
	PL	X1,SCD	.SENSE CONTINUOUS END POINTS	ROFF	2795
	SX2	400B		ROFF	2796
	LX3	73B		ROFF	2797
	3X6	X6+X2		ROFF	2798
	PL	X3,SCD	.SENSE N EVEN	ROFF	2799
	SB4	B4-B7		ROFF	2800
SCP.	LX1	1		ROFF	2801
	PL	X1,SCP1	.SENSE LOW INTENSITY	ROFF	2802
	IX6	X6+X0		ROFF	2803
SCP1	LX6	60B		ROFF	2804
	SA6	A0-B7		ROFF	2805
	RJ	SCAL		ROFF	2806
	SA4	B3		ROFF	2807
	ZR	X4,P10F	.IPEN = 0 DO NOT DRAW	ROFF	2808
	SA3	A0-B7		ROFF	2809
	LX1	30B		ROFF	2810
	3X6	X1+X3		ROFF	2811
	LT	B0,B4,SCP3		ROFF	2812
	SA4	IXCUR		ROFF	2813
	AX6	60B		ROFF	2814
	SA5	A4+B7		ROFF	2815
	LX6	60B		ROFF	2816
	AX1	30B		ROFF	2817
	LX4	44B		ROFF	2818
	3X6	X6+X1		ROFF	2819
	LX5	30B		ROFF	2820
	IX4	X4+X5		ROFF	2821
	IX6	X4+X6		ROFF	2822
	SA6	A0-B7	.OUTPUT VECTOR FROM CURRENT POSITION	ROFF	2823
SCP2	SX7	B7		ROFF	2824
	RJ	POUT		ROFF	2825
	EQ	B0,B0,P10F		ROFF	2826
				ROFF	2827

SCP3	SA6	A0-B7		ROFF	2828
	RJ	SCAL		ROFF	2829
	SA3	A0-B7		ROFF	2830
	3A6	X3+X1		ROFF	2831
	SA6	A0-B7	.STORE FIRST SET END POINTS	ROFF	2832
	EQ	B0,B4,SCP2		ROFF	2833
	RJ	SCAL		ROFF	2834
	LA1	448		ROFF	2835
	3X6	X1		ROFF	2836
	SA6	A0	.STORE SECOND SET END POINTS	ROFF	2837
	EQ	B0,B4,SCP4		ROFF	2838
	RJ	SCAL		ROFF	2839
	SA4	A0		ROFF	2840
	LA1	148		ROFF	2841
	3X6	X4+X1		ROFF	2842
	SA6	A4	.STORE THIRD SET DATA POINTS	ROFF	2843
	EQ	B0,B4,SCP4		ROFF	2844
	RJ	SCAL		ROFF	2845
	SA3	A0		ROFF	2846
	4X2	608		ROFF	2847
	LA1	608		ROFF	2848
	3X7	X1*X2		ROFF	2849
	3X6	-X2*X1		ROFF	2850
	IX6	X3+X6		ROFF	2851
	SA7	A0-B7	.STORE FOURTH SET END POINTS	ROFF	2852
	SA6	A3		ROFF	2853
	EQ	B0,B4,SCP5		ROFF	2854
	RJ	SCAL		ROFF	2855
	SA3	A0-B7		ROFF	2856
	LA1	308		ROFF	2857
	3X6	X1+X3		ROFF	2858
	SA6	A3	.STORE FIFTH SET END POINTS	ROFF	2859
	EQ	B0,B4,SCP5		ROFF	2860
	RJ	SCAL		ROFF	2861
	SA3	A0-B7		ROFF	2862
	3X6	X1+X3		ROFF	2863
	SA6	A3	.STORE SIXTH SET END POINTS	ROFF	2864
	EQ	B0,B0,SCP5		ROFF	2865
SCP4	3X7	B7-B7		ROFF	2866
	RJ	POUT		ROFF	2867
	EQ	B0,B0,P10F		ROFF	2868
SCP5	3A7	3		ROFF	2869
	RJ	POUT		ROFF	2870
	EQ	B0,B0,P10F		ROFF	2871
*	SCAL	SCALING ROUTINE		ROFF	2872
	3SS	1		ROFF	2873
	SA1	B5		ROFF	2874
	ZR	X1,SCL1		ROFF	2875
	IX2	X1-X0		ROFF	2876
	ZR	X2,SCL8		ROFF	2877
	IX3	X2-X0		ROFF	2878
	ZR	X3,SCL9		ROFF	2879
	3X1	X1-15		ROFF	2880
	ZR	X1,SCL9		ROFF	2881
SCL1	SA3	XMIN	.XPOS=(A-XMIN)*XSCALE+XORIG	ROFF	2882
	SA5	B1	.YPOS=(B-YMIN)*YSCALE+YORIG	ROFF	2883
	RX4	X5-X3		ROFF	2884



	SA1	XSCALE		ROFF	2885
	SA5	YMIN		ROFF	2886
	VX0	B0,X4		ROFF	2887
	SA2	B2		ROFF	2888
	RX0	X0*X1		ROFF	2889
	RX5	X2-X5		ROFF	2890
	SA3	XMI		ROFF	2891
	SA4	A1+B7		ROFF	2892
	NX5	B0,X5		ROFF	2893
	RX0	X0+X3		ROFF	2894
	SA2	YMI		ROFF	2895
	NX7	B0,X0		ROFF	2896
	RX5	X5*X4		ROFF	2897
	RX0	X5+X2		ROFF	2898
	VX6	B0,X0		ROFF	2899
	SX0	B7		ROFF	2900
SCL2	JX6	B6,X6	.IXPOS=XPOS=X7	ROFF	2901
	VX3	0		ROFF	2902
	LX6	B6,X6	.IYPOS=YPOS=X6	ROFF	2903
	JX7	B6,X7		ROFF	2904
	IX6	X6+X3		ROFF	2905
	-X7	B6,X7		ROFF	2906
	IX7	X7+X3		ROFF	2907
	ZR	X7,SC-4	.JUMP IXPOS=0	ROFF	2908
	PL	X7,SC-3	.JUMP IXPOS=+	ROFF	2909
	VX7	0		ROFF	2910
SCL3	ZR	30,SC-4		ROFF	2911
	SX3	B0+1024		ROFF	2912
	IX4	X7-X3		ROFF	2913
	VX6	X4,SC-4		ROFF	2914
SCL4	SA7	1023	.IXPOS	ROFF	2915
	SA7	IXPOS		ROFF	2916
	RX3	B0,X7		ROFF	2917
	VX7	B0,X3		ROFF	2918
	SA7	XPOS	.XPOS	ROFF	2919
	ZR	X6,SC-6	.JUMP IYPOS=0	ROFF	2920
	PL	X6,SC-5	.JUMP IYPOS=+	ROFF	2921
	VX6	0		ROFF	2922
SCL5	ZR	30,SC-6		ROFF	2923
	SX3	1024		ROFF	2924
	IX4	X6-X3		ROFF	2925
	VX6	X4,SC-6		ROFF	2926
SCL6	SX6	1023	.IYPOS	ROFF	2927
	SA6	IYPOS		ROFF	2928
	SB1	B1+B7		ROFF	2929
	RX3	B0,X6		ROFF	2930
	SB2	B2+B7		ROFF	2931
	SA1	IXPOS		ROFF	2932
	VX6	B0,X3		ROFF	2933
	SA2	A1+B7	.YPOS	ROFF	2934
	SA6	YPOS		ROFF	2935
	-X1	148		ROFF	2936
	SB4	B4-B7		ROFF	2937
	OX1	X1+X2		ROFF	2938
	VE	B0,B4,SCAL		ROFF	2939
	MX5	1		ROFF	2940
	VZ	X2,SCAL		ROFF	2941

	IX5	138		ROFF	2942
	IX1	X1+X5		ROFF	2943
	EQ	B0,B0,SCAL		ROFF	2944
*	JIF=	DIFFERENTIAL ABSOLUTE SCALING		ROFF	2945
SCL5	SA3	XCUR	.XPOS=A+XCUR	ROFF	2946
	SA2	B1	.YPOS=B+YCUR	ROFF	2947
	IX4	X2+X3		ROFF	2948
	SA1	A3+37		ROFF	2949
	IX7	B0,A4		ROFF	2950
	SA2	B2		ROFF	2951
	IX0	X2+X1		ROFF	2952
	IX6	B0,X0		ROFF	2953
	ZR	B0,SCL2	.JUMP WITH X6=YPOS AND X7=XPOS	ROFF	2954
*	JIF=	ABSOLUTE SCALING		ROFF	2955
SCL9	SA1	B1	.XPOS=X	ROFF	2956
	SA2	B2	.YPOS=Y	ROFF	2957
	IX6	X2		ROFF	2958
	IX7	X1		ROFF	2959
	ZR	B0,SCL2	.JUMP WITH X6=YPOS AND X7=XPOS	ROFF	2960
*	JIF=	UPDATE CURRENT X AND Y POSITIONS		ROFF	2961
PI0F	SA1	XPOS	.XCUR=XPOS	ROFF	2962
	SA3	IXPOS	.IXCUR=IXPOS	ROFF	2963
	IX6	X1		ROFF	2964
	SA2	A1+B7	.YCUR=YPOS	ROFF	2965
	SA4	A3+B7	.IYCUR=IYPOS	ROFF	2966
	IX7	X2		ROFF	2967
	SA6	XCUR		ROFF	2968
	SA7	A6+B7		ROFF	2969
	IX6	X3		ROFF	2970
	IX7	X4		ROFF	2971
	SA6	IXCUR		ROFF	2972
	SA7	A6+B7		ROFF	2973
	EQ	PLOT1	.EXIT	ROFF	2974
*	JIF=	CONTINUOUS PLOT SYMBOL	A,B = POINTS I=NO OF POINTS J=0	ROFF	2975
PSY4	SA2	B3		ROFF	2976
	SA1	B4		ROFF	2977
	IX6	X2,PS3		ROFF	2978
	SB4	X2		ROFF	2979
	SB5	B0		ROFF	2980
	IX6	368		ROFF	2981
	SA6	B368		ROFF	2982
	IX	08CD		ROFF	2983
	IX7	208		ROFF	2984
	SA7	B368		ROFF	2985
	MX1	6		ROFF	2986
	IX6	X1+X6		ROFF	2987
	IX6	6		ROFF	2988
	SB3	X6		ROFF	2989
	EQ	B0,B4,PLOT1	.EXIT N = 0	ROFF	2990
	IX	SCAL		ROFF	2991
	IX5	20008		ROFF	2992
	SA3	WORD1		ROFF	2993
	IX2	-X5*X1		ROFF	2994
	SA4	WORD1+2		ROFF	2995
	IX2	308		ROFF	2996
	IX3	608		ROFF	2997
	IX1	608		ROFF	2998

	3X6	X2+X3		ROFF	2999
	3X4	X1+X3		ROFF	3000
	LX4	148		ROFF	3001
	4X5	608		ROFF	3002
	IX6	X6+X4		ROFF	3003
	3X7	X1*X5		ROFF	3004
	SA7	A0		ROFF	3005
	3X5	-X5*X1		ROFF	3006
	IX6	X6+X5		ROFF	3007
	SA6	A0-B7	.STORE FIRST SET	ROFF	3008
	EQ	B4,B0,PS1		ROFF	3009
	RJ	SCAL		ROFF	3010
	SA3	A0		ROFF	3011
	LX1	308		ROFF	3012
	3X6	X1+X3		ROFF	3013
	SA6	A0	.STORE SECOND SET	ROFF	3014
	EQ	B4,B0,PS1		ROFF	3015
	RJ	SCAL		ROFF	3016
	SA3	A0		ROFF	3017
	3X6	X1+X3		ROFF	3018
	SA6	A0	.STORE THIRD SET	ROFF	3019
	EQ	B0,B4,PS1		ROFF	3020
	RJ	SCAL		ROFF	3021
	LX1	448		ROFF	3022
	3X6	X1		ROFF	3023
	SA6	A0+B7	.STORE FOURTH SET	ROFF	3024
	EQ	B0,B4,PS2		ROFF	3025
	RJ	SCAL		ROFF	3026
	SA3	A0+B7		ROFF	3027
	LX1	148		ROFF	3028
	3X6	X1+X3		ROFF	3029
	SA6	A0+B7	.STORE FIFTH SET	ROFF	3030
	EQ	B0,B0,PS2		ROFF	3031
PS1	3X7	B7+B7		ROFF	3032
	RJ	POUT		ROFF	3033
	EQ	B0,B0,P10F		ROFF	3034
PS2	3X7	3		ROFF	3035
	RJ	POUT		ROFF	3036
	EQ	B0,B0,P10F		ROFF	3037
*	.SET SYMBOL PARAMETERS	A=0,B=ORIENT,I=-1,J=ISIZ,K=5		ROFF	3038
PS3	SA3	B2		ROFF	3039
	LX1	7		ROFF	3040
	SA4	WORD1		ROFF	3041
	4X6	0		ROFF	3042
	ZR	X3,PS4		ROFF	3043
	SA6	B7+B7		ROFF	3044
PS4	SA5	WORD1+2		ROFF	3045
	3X6	X6+X4		ROFF	3046
	IX7	X5 (1		ROFF	3047
	SA6	A4		ROFF	3048
	SA7	A5		ROFF	3049
	EQ	PLOT1	.EXIT	ROFF	3050
*	.PLOT 1-20 CHARACTERS IN TAB MODE			ROFF	3051
*	.A=ADDR OF STRING B = ORIENT I = NO OF CHAR J = SIZE K = 6			ROFF	3052
PC	SA1	B3	.READ N	ROFF	3053
	SA2	B4	.READ CHAR SIZE	ROFF	3054
	4X3	58		ROFF	3055

## AFWL-TR-72-139

	3X2	-X3*X2	
	ZR	X1,PLJF1	.EXIT N = 0
	3X3	X2	
	-X2	7	
	3X6	X3	
	-X6	3	
	5X5	X3-2	
	PL	X5,PC1	
	5X6	108	
	ZR	X3,PC1	
PC1	5X6	148	
	SA1	82	.READ ORIENTATION PARAMETER
	4X7	0	
	ZR	X1,PC2	
	5X1	87+87	
	3X7	X6	
PC2	4X6	0	
	5A6	IX	
	5A3	WORD1+1	
	5X4	17778	
	IX6	X1+X3	
	5A1	IXCUR	
	5A3	A7+87	
	3X6	X6+X2	
	LX6	608	
	3X1	X1*X4	
	5A7	A6+87	
	LX1	448	
	3X3	X3*X4	
	-X3	308	
	IX3	X1+X3	
	3X6	X3+X6	
	5A1	81	
	5Ac	A0-87	
	RJ	08C0	
	5A2	83	
	5A6	TEMP	
	5X7	X2-123	
	NS	X7,PC3	
	5A1	81+87	
	RJ	08C0	
	5A6	A6+87	
PC3	5A2	83	
	5A5	TEMP	
	5B5	X2	.B5=N
	IX2	X2+X0	
	5A3	A0-87	
	4X7	87,X2	
	3X1	X5	
	5B6	X7	.B6 = N/2
	4X4	148	.X4 = MASK
	5X7	85	
	5B3	87	
	5X5	56028	.X5 = TAB MODE EXIT
	9X2	X4*X1	
	-X1	148	
	LX2	308	

ROFF	3056
ROFF	3057
ROFF	3058
ROFF	3059
ROFF	3060
ROFF	3061
ROFF	3062
ROFF	3063
ROFF	3064
ROFF	3065
ROFF	3066
ROFF	3067
ROFF	3068
ROFF	3069
ROFF	3070
ROFF	3071
ROFF	3072
ROFF	3073
ROFF	3074
ROFF	3075
ROFF	3076
ROFF	3077
ROFF	3078
ROFF	3079
ROFF	3080
ROFF	3081
ROFF	3082
ROFF	3083
ROFF	3084
ROFF	3085
ROFF	3086
ROFF	3087
ROFF	3088
ROFF	3089
ROFF	3090
ROFF	3091
ROFF	3092
ROFF	3093
ROFF	3094
ROFF	3095
ROFF	3096
ROFF	3097
ROFF	3098
ROFF	3099
ROFF	3100
ROFF	3101
ROFF	3102
ROFF	3103
ROFF	3104
ROFF	3105
ROFF	3106
ROFF	3107
ROFF	3108
ROFF	3109
ROFF	3110
ROFF	3111
ROFF	3112

DX6	X3+X2		
SB6	B6-B7		
SB4	B0	.B4 = SHIFT COUNT FOR X5	
EQ	B0,B6,PC11	.JUMP N = 2	
DX2	X1*X4		
SB3	B3+B7		
LX1	148		
-X2	148		
DX6	X6+X2		
SA6	A0-B7		
SB4	608		
SB6	B6-B7		
YX6	0		
EQ	B6,B0,PC7	.JUMP N = 4	
DX6	X4*X1		
SB6	B6-B7		
LX1	148		
SB4	448		
EQ	B0,B6,PC7	.JUMP N = 6	
DX2	X1*X4		
LX1	148		
LA2	608		
DX6	X6+X2		
SB6	B6-B7		
SB4	308		
EQ	B0,B6,PC7	.JUMP N = 8	
DX2	X1*X4		
SB4	148		
SB6	B6-B7		
LX2	448		
DX6	X2+X6		
EQ	B0,B6,PC7	.JUMP N = 10	
SA1	A5+B7	.READ NEXT WORD OF CHARACTERS	
SB4	B0		
DX2	X1*X4		
SB6	B6-B7		
LX1	148		
-X2	308		
DX6	X6+X2		
EQ	B0,B6,PC7	.JUMP N = 12	
DX2	X1*X4		
LX1	148		
SB6	B6-B7		
LX2	148		
SB4	608		
DX6	X2+X6		
SA6	A0	.STORE SECOND WORD	
YX6	0		
EQ	B0,B6,PC10	.JUMP N = 14	
DX6	X1*X4		
SB4	448		
LX1	148		
SB6	B6-B7		
EQ	B0,B6,PC10	.JUMP N = 16	
DX2	X1*X4		
SB4	308		
-X2	608		
ROFF			3113
ROFF			3114
ROFF			3115
ROFF			3116
ROFF			3117
ROFF			3118
ROFF			3119
ROFF			3120
ROFF			3121
ROFF			3122
ROFF			3123
ROFF			3124
ROFF			3125
ROFF			3126
ROFF			3127
ROFF			3128
ROFF			3129
ROFF			3130
ROFF			3131
ROFF			3132
ROFF			3133
ROFF			3134
ROFF			3135
ROFF			3136
ROFF			3137
ROFF			3138
ROFF			3139
ROFF			3140
ROFF			3141
ROFF			3142
ROFF			3143
ROFF			3144
ROFF			3145
ROFF			3146
ROFF			3147
ROFF			3148
ROFF			3149
ROFF			3150
ROFF			3151
ROFF			3152
ROFF			3153
ROFF			3154
ROFF			3155
ROFF			3156
ROFF			3157
ROFF			3158
ROFF			3159
ROFF			3160
ROFF			3161
ROFF			3162
ROFF			3163
ROFF			3164
ROFF			3165
ROFF			3166
ROFF			3167
ROFF			3168
ROFF			3169

	SB6	B6-B7		ROFF	3170
	LX1	148		ROFF	3171
	BX6	X6+X2		ROFF	3172
	EQ	B0,B6,PC10	.JUMP N = 18	ROFF	3173
	SA2	X1*X4		ROFF	3174
	LX2	448		ROFF	3175
	BX6	X2+X6		ROFF	3176
	SB4	148		ROFF	3177
PC5	EQ	B0,B0,PC10	.N = 20	ROFF	3178
	SA4	IX	.UPDATE CURRENT X, Y POSITIONS	ROFF	3179
	SX2	B5		ROFF	3180
	PX4	B0,X4		ROFF	3181
	SA3	IXCUR		ROFF	3182
	PA2	B0,X2		ROFF	3183
	SA1	A4+B7		ROFF	3184
	JX4	X4*X2		ROFF	3185
	SA5	A3+B7		ROFF	3186
	JA4	B0,X4		ROFF	3187
	PX1	B0,X1		ROFF	3188
	IX6	X4+X3		ROFF	3189
	SA6	A3		ROFF	3190
	JA1	X2*X1		ROFF	3191
	PX6	B0,X6		ROFF	3192
	JX6	B0,X6		ROFF	3193
	JX1	B0,X1		ROFF	3194
	IA7	X5+X1		ROFF	3195
	SA7	A6+B7		ROFF	3196
	PX7	B0,X7		ROFF	3197
	SA6	XCUR		ROFF	3198
	VX7	B0,X7		ROFF	3199
	SA7	A6+B7		ROFF	3200
PC7	EQ	PLOT1	.EXIT	ROFF	3201
	LX5	B4,X5		ROFF	3202
	SX7	B3		ROFF	3203
	SA6	X6+X5		ROFF	3204
	SA6	A0		ROFF	3205
	SA0	DATA+1		ROFF	3206
	QJ	POUT		ROFF	3207
PC10	EQ	B0,B0,PC5		ROFF	3208
	SA0	A0+B7	.N = 14,16,18,20	ROFF	3209
	SB3	B3+B7		ROFF	3210
PC11	EQ	B0,B0,PC7		ROFF	3211
	SA6	A0-B7	.N = 2	ROFF	3212
	EQ	B0,B0,PC7		ROFF	3213
	.DISPLAY CODE TO EXTERNAL BCD CONVERSION			ROFF	3214
DBCU	JSS	1		ROFF	3215
	VX6	0		ROFF	3216
	SA2	10		ROFF	3217
	VX3	6		ROFF	3218
UJ1	JX4	X1*X3	.CONVERT TO EXTERNAL BCD	ROFF	3219
	LX4	6		ROFF	3220
	SA4	X4+EBD		ROFF	3221
	LX6	6		ROFF	3222
	BX6	X6+X4		ROFF	3223
	LX1	6		ROFF	3224
	IX2	X2-X0		ROFF	3225
	VZ	X2,081		ROFF	3226

	EQ	B0,B0,B0C0		ROFF	3227
*.RECEIVE		INITIAL X AND Y VALUES AND ERROR FLAG-INITIAL GRAPH		ROFF	3228
		.A=XINIT, B=YINIT, I=IERR, J= , K=7		ROFF	3229
PINT	SA1	B1	.READ X INITIAL	ROFF	3230
	SA2	B2	.READ Y INITIAL	ROFF	3231
	SX6	X1		ROFF	3232
	SX7	X2		ROFF	3233
	SA6	XCUR		ROFF	3234
	SA7	A6+B7		ROFF	3235
	JX6	B6,X6		ROFF	3236
	LX6	B6,X6		ROFF	3237
	JX7	B6,X7		ROFF	3238
	LX7	B6,X7		ROFF	3239
	SA6	IXCUR		ROFF	3240
	SB2	BUFFC		ROFF	3241
	SA1	BUFF		ROFF	3242
	SA7	A6+B7		ROFF	3243
	SB2	B0-B2		ROFF	3244
	NZ	X1,PINTB	.JUMP BUFFER POINTER FOUND	ROFF	3245
	RJ	GETBA		ROFF	3246
	LT	B2,B0,PINTA		ROFF	3247
	SX6	B2		ROFF	3248
	SA6	BUFF		ROFF	3249
PINTA	EQ	B0,B0,PINTB		ROFF	3250
	SB2	BUFFD		ROFF	3251
	SB2	B0-B2		ROFF	3252
	RJ	GETBA		ROFF	3253
	LT	B2,B0,PINTC		ROFF	3254
	SX6	B2		ROFF	3255
PINTB	SA6	BUFF		ROFF	3256
	SA1	B3		ROFF	3257
	RJ	X1,PLDT1	.EXIT NO PREAMBLE SUPPRESSION	ROFF	3258
	SX7	B7		ROFF	3259
	SY6	6-00B		ROFF	3260
	LX6	60B		ROFF	3261
	SA6	A0-B7		ROFF	3262
	RJ	POUT	.OUTPUT PREAMBLE SUP CONTROL WORD	ROFF	3263
	EQ	PLOT1	.EXIT	ROFF	3264
PINTC	SX2	MSG		ROFF	3265
	SX6	152307B		ROFF	3266
	LX6	52B		ROFF	3267
OVER	SA1	B7		ROFF	3268
	NZ	X1,OVER		ROFF	3269
	IX6	A6+X2		ROFF	3270
	SA6	B7		ROFF	3271
	RJ	ABNORM.		ROFF	3272
	LT	B0,B4,NAME		ROFF	3273
*.TERMINATE GRAPH-WRITE END OF RECORD				ROFF	3274
PTER	SX5	26B	.X5=STATUS=26B=EOR	ROFF	3275
	RJ	CALL	.PUT UP CIO CALL	ROFF	3276
	EQ	PLOT1	.EXIT	ROFF	3277
SPSM	SA1	B1	.SET PLOT SYMBOL PARAMETERS	ROFF	3278
	SA2	B2		ROFF	3279
	SX5	2003B		ROFF	3280
	SA3	B3		ROFF	3281
	SX6	3001B		ROFF	3282
	SX7	3041B		ROFF	3283

	ZR	X1,SPSM1	.SENSE CASE 0	ROFF	3284
	SX4	1008		ROFF	3285
	IX6	X6+X4		ROFF	3286
	IX7	X7+X4		ROFF	3287
	IX5	X5+X4		ROFF	3288
SPSM1	ZR	X2,SPSM2	.SENSE NO ITALICS	ROFF	3289
	SX4	4		ROFF	3290
	IX6	X6+X4		ROFF	3291
	IX7	X7+X4		ROFF	3292
SPSM2	ZR	X3,SPSM3	.SENSE HIGH INTENSITY	ROFF	3293
	IX6	X6-X0	.REMOVE HIGH INTENSITY BIT	ROFF	3294
	IX7	X7-X0		ROFF	3295
SPSM3	SA6	WORD1	.STORE UPDATED CONTROL WORDS	ROFF	3296
	IX6	X5		ROFF	3297
	SA7	A6+87		ROFF	3298
	SA6	A7+87		ROFF	3299
	EQ	PLOT1	.GO HOME	ROFF	3300
* POUT	.STORE	X6 IN FILMPL BUFFER AND ADVANCE IN		ROFF	3301
	PS			ROFF	3302
	SA2	A0-87		ROFF	3303
	SA1	BUFF		ROFF	3304
	IX7	4E		ROFF	3305
	SB6	X1		ROFF	3306
	IX6	X2+X7		ROFF	3307
	AX7	4E		ROFF	3308
PA6	SA3	8E+2	.READ IN	ROFF	3309
	SA6	X3	.STORE DATA WORD AT IN	ROFF	3310
POUT1	SX3	X3+37	.INCREMENT IN	ROFF	3311
	SA5	8E+4	.READ LIMIT	ROFF	3312
	SA5	X5		ROFF	3313
	IX5	X5-X3		ROFF	3314
	VZ	X5,POJT2	.JUMP IN NOT LIMIT	ROFF	3315
	SA3	8E+87	.SET IN = FIRST	ROFF	3316
	SX3	X3		ROFF	3317
POUT2	SA5	A5-87	.READ OUT	ROFF	3318
	IX5	X5-X3		ROFF	3319
	ZR	X5,POJT5	.JUMP TO DUMP BUFFER (IN+1=OUT)	ROFF	3320
	IX6	X3		ROFF	3321
	SA6	A5-87	.STORE UPDATED IN	ROFF	3322
	ZR	80,POJT6		ROFF	3323
POUT5	SX5	168	.DUMP BUFFER (BUFFERED I/O)	ROFF	3324
	RJ	CALL	.PUT UP CIO CALL	ROFF	3325
	SA3	8E+2	.READ IN	ROFF	3326
	ZR	80,POUT1		ROFF	3327
POUT6	IX7	X7-X0		ROFF	3328
	NG	X7,POJT		ROFF	3329
	ZL	X7,POJT		ROFF	3330
	SA2	A0		ROFF	3331
	SA0	A0+87		ROFF	3332
	IX6	X2		ROFF	3333
	EQ	80,80,PAG		ROFF	3334
* CALL	.PUT UP	CIO CALL		ROFF	3335
	. X5=	BUFFER OPERATION		ROFF	3336
	PS			ROFF	3337
	SA2	BUFF		ROFF	3338
	SA3	031117B		ROFF	3339
	IX6	42		ROFF	3340



AFWL-TR-72-139

	SA1	X2		ROFF	3341	
	AX3	528		ROFF	3342	
	AX6	X1*X6		ROFF	3343	
	IX6	X6+X5		ROFF	3344	
	IX4	X3+X2		ROFF	3345	
	SA6	A1	.STORE BA	ROFF	3346	
CALL1	AX6	X4		ROFF	3347	
	SA2	B7		ROFF	3348	
	NZ	X2,CALL1		ROFF	3349	
CALL2	SA6	B7	.CALL CALL	ROFF	3350	
	RJ	XRGCL		ROFF	3351	
	SA1	A1		ROFF	3352	
	IX1	59		ROFF	3353	
	IX1	X1,CALL		ROFF	3354	
WORD1		50,80,CALL2		ROFF	3355	
		32018		ROFF	3356	
		30418		ROFF	3357	
		20008		ROFF	3358	
BUFFC	CON	06111000014551617248		ROFF	3359	
BUFFD	CON	15063000000000000000		ROFF	3360	
MSG	CON	06111000014551617248		ROFF	3361	
	CON	550400032401220504558		ROFF	3362	
	DATA	0		ROFF	3363	
	.TABLE FOR DISPLAY TO EXTERNAL BCD CONVERSION				ROFF	3364
		EXTERNAL	DISPLAY	ROFF	3365	
2300	CON	208	.SPACE	ROFF	3366	
	CON	618	.A	ROFF	3367	
	CON	628	.B	ROFF	3368	
	CON	638	.C	ROFF	3369	
	CON	648	.D	ROFF	3370	
	CON	658	.E	ROFF	3371	
	CON	668	.F	ROFF	3372	
	CON	678	.G	ROFF	3373	
	CON	708	.H	ROFF	3374	
	CON	718	.I	ROFF	3375	
	CON	418	.J	ROFF	3376	
	CON	428	.K	ROFF	3377	
	CON	438	.L	ROFF	3378	
	CON	448	.M	ROFF	3379	
	CON	458	.N	ROFF	3380	
	CON	468	.O	ROFF	3381	
	CON	478	.P	ROFF	3382	
	CON	508	.Q	ROFF	3383	
	CON	518	.R	ROFF	3384	
	CON	228	.S	ROFF	3385	
	CON	238	.T	ROFF	3386	
	CON	248	.U	ROFF	3387	
	CON	258	.V	ROFF	3388	
	CON	268	.W	ROFF	3389	
	CON	278	.X	ROFF	3390	
	CON	308	.Y	ROFF	3391	
	CON	318	.Z	ROFF	3392	
	CON	128	.ZERO	ROFF	3393	
	CON	018	.1	ROFF	3394	
	CON	028	.2	ROFF	3395	
	CON	038	.3	ROFF	3396	
	CON	048	.4	ROFF	3397	

AFWL-TR-72-139

	CON	058	.5		ROFF	3398
	CON	068	.6		ROFF	3399
	CON	078	.7		ROFF	3400
	CON	08	.8		ROFF	3401
	CON	118	.9		ROFF	3402
	CON	608	+		ROFF	3403
	CON	408	-		ROFF	3404
	CON	548	*		ROFF	3405
	CON	218	/		ROFF	3406
	CON	348	(		ROFF	3407
	CON	748	)		ROFF	3408
3368	CON	208			ROFF	3409
	CON	138	=		ROFF	3410
	CON	208	SPACE		ROFF	3411
	CON	338	,		ROFF	3412
	CON	738	:		ROFF	3413
	CON	008	.		ROFF	3414
	CON	158	.		ROFF	3415
	CON	168	.		ROFF	3416
	CON	178	.		ROFF	3417
	CON	328	.		ROFF	3418
	CON	358	.		ROFF	3419
	CON	148	.		ROFF	3420
	CON	378	.		ROFF	3421
	CON	528	.		ROFF	3422
	CON	558	.		ROFF	3423
	CON	538	.		ROFF	3424
	CON	578	.		ROFF	3425
	CON	728	.		ROFF	3426
	CON	758	.		ROFF	3427
	CON	768	.		ROFF	3428
	CON	778	.		ROFF	3429
XPOS	8SSZ	1			ROFF	3430
YPOS	8SSZ	1			ROFF	3431
XCUR	8SSZ	1			ROFF	3432
YCUR	8SSZ	1			ROFF	3433
IXPOS	8SSZ	1			ROFF	3434
IYPOS	8SSZ	1			ROFF	3435
IXCUR	8SSZ	1			ROFF	3436
IYCUR	8SSZ	1			ROFF	3437
DATA	8SSZ	3			ROFF	3438
ORIEN	8SSZ	1			ROFF	3439
IX	8SSZ	1			ROFF	3440
IY	8SSZ	1			ROFF	3441
TEMP	8SSZ	2			ROFF	3442
BUFF	8SSZ	1			ROFF	3443
SAVAD	8SSZ	1			ROFF	3444
END					ROFF	3445

AFWL-TR-72-139

TRANSLATE TABLE EBCDIC TO LITTON CODE FOR MTST

SUBROUTINE WRT9209 (ICG,LINE,LEN)

MTST READS FORTRAN OUTPUT FILE ON TAPE1 AND WRITES MTST CODES  
ON TAPE2 FOR CONVERSION VIA THE LITTON TAPE/MTST UNIT.

PROGRAM BY HARRY M. MURPHY, JR., 4 FEBRUARY 1972.  
MODIFIED 17F-372 TO PERMIT WRITING MULTIPLE MTST CARTRIDGES.  
REVISED INTO SUBROUTINE  
BY LT. CLIFFORD E. RHOADES, JR. 25 FEBRUARY

COMMON LWD(3700)

COMMON /BLK1/ LBY,LWP

COMMON /GO/ ITRZ,ITR(255)  
COMMON /PAGES/ IPAGES,MES(5)

DIMENSION KOL(10), LINE(135)

LOGICAL DONE,SECOND,BKSL,GREEK

DATA KOL/16318,16048,16408,16448,16348,16208,16608,16248,16648,16  
1703/

DATA DONE,SECOND,BKSL,GREEK/.FALSE.,.FALSE.,.FALSE.,.FALSE./

DATA KPLS,KONE,KZRO/1H+,1H1,1H0/  
DATA KBL/64/

DATA MCRC,MFC),MSTC,MSTX/16108,16528,16548,16578/  
DATA MBSP,M8.K/14158,16138/

DATA ITRZ/145114328/  
DATA ITR(1)/145414658/  
DATA ITR(2)/145414728/  
DATA ITR(3)/145414668/  
DATA ITR(4)/145414268/  
DATA ITR(5)/145118/  
DATA ITR(6)/145414328/  
DATA ITR(7)/145414078/  
DATA ITR(8)/145414138/  
DATA ITR(9)/145414618/  
DATA ITR(10)/145414018/  
DATA ITR(11)/145416018/  
DATA ITR(12)/145414418/  
DATA ITR(13)/16138/  
DATA ITR(14)/145414378/  
DATA ITR(15)/145414658/  
DATA ITR(16)/145414528/  
DATA ITR(17)/145414428/

ROFF 3446  
ROFF 3447  
ROFF 3448  
ROFF 3449  
ROFF 3450  
ROFF 3451  
ROFF 3452  
ROFF 3453  
ROFF 3454  
ROFF 3455  
ROFF 3456  
ROFF 3457  
ROFF 3458  
ROFF 3459  
ROFF 3460  
ROFF 3461  
ROFF 3462  
ROFF 3463  
ROFF 3464  
ROFF 3465  
ROFF 3466  
ROFF 3467  
ROFF 3468  
ROFF 3469  
ROFF 3470  
ROFF 3471  
ROFF 3472  
ROFF 3473  
ROFF 3474  
ROFF 3475  
ROFF 3476  
ROFF 3477  
ROFF 3478  
ROFF 3479  
ROFF 3480  
ROFF 3481  
ROFF 3482  
ROFF 3483  
ROFF 3484  
ROFF 3485  
ROFF 3486  
ROFF 3487  
ROFF 3488  
ROFF 3489  
ROFF 3490  
ROFF 3491  
ROFF 3492  
ROFF 3493  
ROFF 3494  
ROFF 3495  
ROFF 3496  
ROFF 3497  
ROFF 3498  
ROFF 3499  
ROFF 3500

AFWL-TR-72-139

SUBROUTINE WRT9209

DATA ITR(19)/145414628/  
DATA ITR(19)/145414368/  
DATA ITR(20)/145414053/  
DATA ITR(21)/145414528/  
DATA ITR(22)/16159/  
DATA ITR(23)/145414238/  
DATA ITR(24)/145414238/  
DATA ITR(25)/145414258/  
DATA ITR(26)/145414438/  
DATA ITR(27)/145416438/  
DATA ITR(28)/145414378/  
DATA ITR(29)/145414358/  
DATA ITR(30)/145414528/  
DATA ITR(31)/145414008/  
DATA ITR(32)/145414208/  
DATA ITR(33)/145414228/  
DATA ITR(34)/145414318/  
DATA ITR(35)/145414028/  
DATA ITR(36)/145414468/  
DATA ITR(37)/145414458/  
DATA ITR(38)/145414718/  
DATA ITR(39)/145414068/  
DATA ITR(40)/145414338/  
DATA ITR(41)/145414008/  
DATA ITR(42)/145414528/  
DATA ITR(43)/145414258/  
DATA ITR(44)/145416058/  
DATA ITR(45)/145414529/  
DATA ITR(46)/145414048/  
DATA ITR(47)/145414528/  
DATA ITR(48)/145414529/  
DATA ITR(49)/145414528/  
DATA ITR(50)/145414528/  
DATA ITR(51)/145414528/  
DATA ITR(52)/145414528/  
DATA ITR(53)/145414528/  
DATA ITR(54)/145414528/  
DATA ITR(55)/145414528/  
DATA ITR(56)/145414528/  
DATA ITR(57)/145414528/  
DATA ITR(58)/145416723/  
DATA ITR(59)/145414528/  
DATA ITR(60)/145414529/  
DATA ITR(61)/145414528/  
DATA ITR(62)/145416718/  
DATA ITR(63)/145414528/  
DATA ITR(64)/000016138/  
DATA ITR(65)/145416658/  
DATA ITR(66)/145416728/  
DATA ITR(67)/145416668/  
DATA ITR(68)/145416268/  
DATA ITR(69)/145416228/  
DATA ITR(70)/145416478/  
DATA ITR(71)/145416378/  
DATA ITR(72)/145416328/

ROFF 3501  
ROFF 3502  
ROFF 3503  
ROFF 3504  
ROFF 3505  
ROFF 3506  
ROFF 3507  
ROFF 3508  
ROFF 3509  
ROFF 3510  
ROFF 3511  
ROFF 3512  
ROFF 3513  
ROFF 3514  
ROFF 3515  
ROFF 3516  
ROFF 3517  
ROFF 3518  
ROFF 3519  
ROFF 3520  
ROFF 3521  
ROFF 3522  
ROFF 3523  
ROFF 3524  
ROFF 3525  
ROFF 3526  
ROFF 3527  
ROFF 3528  
ROFF 3529  
ROFF 3530  
ROFF 3531  
ROFF 3532  
ROFF 3533  
ROFF 3534  
ROFF 3535  
ROFF 3536  
ROFF 3537  
ROFF 3538  
ROFF 3539  
ROFF 3540  
ROFF 3541  
ROFF 3542  
ROFF 3543  
ROFF 3544  
ROFF 3545  
ROFF 3546  
ROFF 3547  
ROFF 3548  
ROFF 3549  
ROFF 3550  
ROFF 3551  
ROFF 3552  
ROFF 3553  
ROFF 3554  
ROFF 3555

Reproduced from  
best available copy.

AFWL-TR-72-139

SUBROUTINE WRT9209

DATA ITR(073)/145416618/	ROFF	3556
DATA ITR(074)/000014608/	ROFF	3557
DATA ITR(075)/000014418/	ROFF	3558
DATA ITR(076)/145414478/	ROFF	3559
DATA ITR(077)/000014708/	ROFF	3560
DATA ITR(078)/000014438/	ROFF	3561
DATA ITR(079)/145414408/	ROFF	3562
DATA ITR(080)/000014248/	ROFF	3563
DATA ITR(081)/145416038/	ROFF	3564
DATA ITR(082)/145416628/	ROFF	3565
DATA ITR(083)/145416368/	ROFF	3566
DATA ITR(084)/145416058/	ROFF	3567
DATA ITR(085)/145416423/	ROFF	3568
DATA ITR(086)/145416358/	ROFF	3569
DATA ITR(087)/145416238/	ROFF	3570
DATA ITR(088)/145416633/	ROFF	3571
DATA ITR(089)/145416258/	ROFF	3572
DATA ITR(090)/000016018/	ROFF	3573
DATA ITR(091)/000014348/	ROFF	3574
DATA ITR(092)/000014648/	ROFF	3575
DATA ITR(093)/000014303/	ROFF	3576
DATA ITR(094)/000016278/	ROFF	3577
DATA ITR(095)/000016738/	ROFF	3578
DATA ITR(096)/000016739/	ROFF	3579
DATA ITR(097)/000016373/	ROFF	3580
DATA ITR(098)/145416318/	ROFF	3581
DATA ITR(099)/145416028/	ROFF	3582
DATA ITR(100)/145416468/	ROFF	3583
DATA ITR(101)/145416458/	ROFF	3584
DATA ITR(102)/145416718/	ROFF	3585
DATA ITR(103)/145416068/	ROFF	3586
DATA ITR(104)/145416338/	ROFF	3587
DATA ITR(105)/145416008/	ROFF	3588
DATA ITR(106)/145414523/	ROFF	3589
DATA ITR(107)/000016678/	ROFF	3590
DATA ITR(108)/000014203/	ROFF	3591
DATA ITR(109)/000014738/	ROFF	3592
DATA ITR(110)/145414038/	ROFF	3593
DATA ITR(111)/000014378/	ROFF	3594
DATA ITR(112)/145414308/	ROFF	3595
DATA ITR(113)/145414048/	ROFF	3596
DATA ITR(114)/145414738/	ROFF	3597
DATA ITR(115)/145416733/	ROFF	3598
DATA ITR(116)/145414703/	ROFF	3599
DATA ITR(117)/145414208/	ROFF	3600
DATA ITR(118)/145414608/	ROFF	3601
DATA ITR(119)/145414243/	ROFF	3602
DATA ITR(120)/145414648/	ROFF	3603
DATA ITR(121)/145414678/	ROFF	3604
DATA ITR(122)/000014278/	ROFF	3605
DATA ITR(123)/000014448/	ROFF	3606
DATA ITR(124)/000014498/	ROFF	3607
DATA ITR(125)/000016218/	ROFF	3608
DATA ITR(126)/000016438/	ROFF	3609
DATA ITR(127)/000014218/	ROFF	3610

## SUBROUTINE WRT3209

DATA ITR(128)/145416218/	ROFF	3611
DATA ITR(129)/000016658/	ROFF	3612
DATA ITR(130)/000016728/	ROFF	3613
DATA ITR(131)/000016668/	ROFF	3614
DATA ITR(132)/000016268/	ROFF	3615
DATA ITR(133)/000016228/	ROFF	3616
DATA ITR(134)/000016478/	ROFF	3617
DATA ITR(135)/000016078/	ROFF	3618
DATA ITR(136)/000016328/	ROFF	3619
DATA ITR(137)/000016618/	ROFF	3620
DATA ITR(138)/145414528/	ROFF	3621
DATA ITR(139)/145414528/	ROFF	3622
DATA ITR(140)/145414528/	ROFF	3623
DATA ITR(141)/145414528/	ROFF	3624
DATA ITR(142)/145414528/	ROFF	3625
DATA ITR(143)/145414528/	ROFF	3626
DATA ITR(144)/145414213/	ROFF	3627
DATA ITR(145)/16338/	ROFF	3628
DATA ITR(146)/16628/	ROFF	3629
DATA ITR(147)/16368/	ROFF	3630
DATA ITR(148)/16358/	ROFF	3631
DATA ITR(149)/16428/	ROFF	3632
DATA ITR(150)/16358/	ROFF	3633
DATA ITR(151)/16238/	ROFF	3634
DATA ITR(152)/16638/	ROFF	3635
DATA ITR(153)/16258/	ROFF	3636
DATA ITR(154)/145414528/	ROFF	3637
DATA ITR(155)/145414528/	ROFF	3638
DATA ITR(156)/145414528/	ROFF	3639
DATA ITR(157)/145414528/	ROFF	3640
DATA ITR(158)/145414528/	ROFF	3641
DATA ITR(159)/145414528/	ROFF	3642
DATA ITR(160)/145414528/	ROFF	3643
DATA ITR(161)/145416418/	ROFF	3644
DATA ITR(162)/16318/	ROFF	3645
DATA ITR(163)/16028/	ROFF	3646
DATA ITR(164)/16468/	ROFF	3647
DATA ITR(165)/16458/	ROFF	3648
DATA ITR(166)/16718/	ROFF	3649
DATA ITR(167)/16368/	ROFF	3650
DATA ITR(168)/16338/	ROFF	3651
DATA ITR(169)/16308/	ROFF	3652
DATA ITR(170)/145414528/	ROFF	3653
DATA ITR(171)/145414528/	ROFF	3654
DATA ITR(172)/145414528/	ROFF	3655
DATA ITR(173)/145414528/	ROFF	3656
DATA ITR(174)/145414528/	ROFF	3657
DATA ITR(175)/145414528/	ROFF	3658
DATA ITR(176)/145416308/	ROFF	3659
DATA ITR(177)/145416048/	ROFF	3660
DATA ITR(178)/145416408/	ROFF	3661
DATA ITR(179)/145416448/	ROFF	3662
DATA ITR(180)/145416348/	ROFF	3663
DATA ITR(181)/145416208/	ROFF	3664
DATA ITR(182)/145416608/	ROFF	3665

AFWL-TR-72-139

SUBROUTINE ART9209

DATA ITR(183)/145416248/	ROFF	3666
DATA ITR(184)/145416648/	ROFF	3667
DATA ITR(185)/145416708/	ROFF	3668
DATA ITR(186)/145416678/	ROFF	3669
	ROFF	3670
DATA ITR(187)/000014048/	ROFF	3671
DATA ITR(188)/145416378/	ROFF	3672
DATA ITR(189)/000016048/	ROFF	3673
DATA ITR(190)/145414278/	ROFF	3674
DATA ITR(191)/145414228/	ROFF	3675
DATA ITR(192)/145414448/	ROFF	3676
DATA ITR(193)/000014658/	ROFF	3677
DATA ITR(194)/000014728/	ROFF	3678
DATA ITR(195)/000014668/	ROFF	3679
DATA ITR(196)/000014268/	ROFF	3680
DATA ITR(197)/000014228/	ROFF	3681
DATA ITR(198)/000014478/	ROFF	3682
DATA ITR(199)/000014078/	ROFF	3683
DATA ITR(200)/000014328/	ROFF	3684
DATA ITR(201)/000014618/	ROFF	3685
DATA ITR(202)/145414528/	ROFF	3686
DATA ITR(203)/145414528/	ROFF	3687
DATA ITR(204)/145414528/	ROFF	3688
DATA ITR(205)/145414528/	ROFF	3689
DATA ITR(206)/145414528/	ROFF	3690
DATA ITR(207)/145414528/	ROFF	3691
DATA ITR(208)/145414348/	ROFF	3692
DATA ITR(209)/000014038/	ROFF	3693
DATA ITR(210)/000014628/	ROFF	3694
DATA ITR(211)/000014368/	ROFF	3695
DATA ITR(212)/000014058/	ROFF	3696
DATA ITR(213)/000014428/	ROFF	3697
DATA ITR(214)/000014358/	ROFF	3698
DATA ITR(215)/000014238/	ROFF	3699
DATA ITR(216)/000014238/	ROFF	3700
DATA ITR(217)/000014258/	ROFF	3701
DATA ITR(218)/145414528/	ROFF	3702
DATA ITR(219)/145414528/	ROFF	3703
DATA ITR(220)/145414528/	ROFF	3704
DATA ITR(221)/145414528/	ROFF	3705
DATA ITR(222)/145414528/	ROFF	3706
DATA ITR(223)/145414528/	ROFF	3707
DATA ITR(224)/145414528/	ROFF	3708
DATA ITR(225)/145416278/	ROFF	3709
DATA ITR(226)/000014318/	ROFF	3710
DATA ITR(227)/000014028/	ROFF	3711
DATA ITR(228)/000014468/	ROFF	3712
DATA ITR(229)/000014458/	ROFF	3713
DATA ITR(230)/000014718/	ROFF	3714
DATA ITR(231)/000014068/	ROFF	3715
DATA ITR(232)/000014338/	ROFF	3716
DATA ITR(233)/000014008/	ROFF	3717
DATA ITR(234)/145414528/	ROFF	3718
DATA ITR(235)/145414528/	ROFF	3719
DATA ITR(236)/145414528/	ROFF	3720

AFWL-TR-72-139

SUBROUTINE MRT9209

DATA ITR(237)/145414528/  
DATA ITR(238)/145414528/  
DATA ITR(239)/145414528/  
DATA ITR(240)/000016308/  
DATA ITR(241)/000016368/  
DATA ITR(242)/000016408/  
DATA ITR(243)/000016448/  
DATA ITR(244)/000016348/  
DATA ITR(245)/000016208/  
DATA ITR(246)/000016608/  
DATA ITR(247)/000016248/  
DATA ITR(248)/000016648/  
DATA ITR(249)/000016708/  
DATA ITR(250)/145114528/  
DATA ITR(251)/145114528/  
DATA ITR(252)/145114528/  
DATA ITR(253)/145114528/  
DATA ITR(254)/145114528/  
DATA ITR(255)/145114528/  
DATA MES/304 NUMBER OF ROFF PAGES PRINTED /  
DATA MES(5)/J/  
DATA IPAGES/0/  
IF (SECOND) GO TO 2

SECOND=.TRUE.

REWIND 9

END FILE 9

DO 1 I=1,15

LWD(I)=165115>11651165116513

LWD(16)=165116511651165115768

LWD(17)=167615761676167616758

LWD(18)=167615761676167616758

LWD(19)=167615761676167616038

LWD(20)=160016001600160016008

LWD(21)=160016001600160016038

LWD(22)=160016001600160016008

LWD(23)=160016001600160016008

LWD(24)=160016001600161014658

LWD(25)=163016301630160416578

NCART=1

NMTST=0

NMTSTS=3

LBY=0

LWP=26

MAIN LOOP STARTS HERE.

BKSL=.FALSE.

OBTAIN OUTPUT LINE AND SEARCH FOR CARRIAGE CONTROL

IF (ICG.EQ.KOYE) GO TO 14

ROFF 3721  
ROFF 3722  
ROFF 3723  
ROFF 3724  
ROFF 3725  
ROFF 3726  
ROFF 3727  
ROFF 3728  
ROFF 3729  
ROFF 3730  
ROFF 3731  
ROFF 3732  
ROFF 3733  
ROFF 3734  
ROFF 3735  
ROFF 3736  
ROFF 3737  
ROFF 3738  
ROFF 3739  
ROFF 3740  
ROFF 3741  
ROFF 3742  
ROFF 3743  
ROFF 3744  
ROFF 3745  
ROFF 3746  
ROFF 3747  
ROFF 3748  
ROFF 3749  
ROFF 3750  
ROFF 3751  
ROFF 3752  
ROFF 3753  
ROFF 3754  
ROFF 3755  
ROFF 3756  
ROFF 3757  
ROFF 3758  
ROFF 3759  
ROFF 3760  
ROFF 3761  
ROFF 3762  
ROFF 3763  
ROFF 3764  
ROFF 3765  
ROFF 3766  
ROFF 3767  
ROFF 3768  
ROFF 3769  
ROFF 3770  
ROFF 3771  
ROFF 3772  
ROFF 3773  
ROFF 3774  
ROFF 3775



AFWL-TR-72-139

	SUBROUTINE WRT9209		
	IF (ICC.EQ.K <sup>2</sup> .S) GO TO 4	ROFF	3776
	IF (ICC.NE.KZ40) GO TO 3	ROFF	3777
1		ROFF	3778
2	HAVE ZERO CARRIAGE CONTROL CHARACTER. WRITE CR AND FEED CODES.	ROFF	3779
3	CALL PACK (MCRG)	ROFF	3780
4	CALL PACK (MFDG)	ROFF	3781
5	CALL PACK (MCRG)	ROFF	3782
6	CALL PACK (MFDG)	ROFF	3783
7	GO TO 5	ROFF	3784
8		ROFF	3785
9		ROFF	3786
10	BACKSPACE WHOLE LINE	ROFF	3787
11		ROFF	3788
12	IOMAX=IMAX	ROFF	3789
13	BKSL=.TRUE.	ROFF	3790
14		ROFF	3791
15	SCAN LINE FOR LAST NON BLANK CHARACTER.	ROFF	3792
16		ROFF	3793
17	CONTINUE	ROFF	3794
18	IF (LEN.LT.1) GO TO 13	ROFF	3795
19	I=LEN+1	ROFF	3796
20	I=I-1	ROFF	3797
21	IF (LINE(I).NE.KBL) GO TO 7	ROFF	3798
22	IF (I-I) 6,13,13	ROFF	3799
23	IMAX=I	ROFF	3800
24	IF (.NOT.BKSL) GO TO 9	ROFF	3801
25	IF (IOMAX.LT.1) GO TO 9	ROFF	3802
26		ROFF	3803
27	DO 8 I=1,IOMAX	ROFF	3804
28	CALL PACK (M3SP)	ROFF	3805
29		ROFF	3806
30	TRANSLATE FROM EBCDIC TO MTST CODE.	ROFF	3807
31		ROFF	3808
32	CONTINUE	ROFF	3809
33	DO 12 I=1,IMAX	ROFF	3810
34	J=LINE(I)	ROFF	3811
35	J=ITRIJ	ROFF	3812
36	IF (J.EQ.MBLK) GO TO 11	ROFF	3813
37	IF (GREEK.AN).J.LT.4096) GO TO 10	ROFF	3814
38	IF (GREEK) GO TO 11	ROFF	3815
39	IF (J.LT.4096) GO TO 11	ROFF	3816
40	CALL PACK (MSTC)	ROFF	3817
41	GREEK=.TRUE.	ROFF	3818
42	GO TO 11	ROFF	3819
43	CALL PACK (MSTC)	ROFF	3820
44	GREEK=.FALSE.	ROFF	3821
45	CALL PACK (J)	ROFF	3822
46	CONTINUE	ROFF	3823
47	CONTINUE	ROFF	3824
48	IF (LMP.LT.3568) RETURN	ROFF	3825
49		ROFF	3826
50	LWD BUFFER FULL.	ROFF	3827
51		ROFF	3828
52	CALL PACK (MCRG)	ROFF	3829
53		ROFF	3830

AFWL-TR-72-139

SUBROUTINE WRT9209

3		ROFF	3831
3	END CURRENT RECORD.	ROFF	3832
3		ROFF	3833
14	IF (LBY.EQ.4) GO TO 15	ROFF	3834
	CALL PACK (MFC)	ROFF	3835
	GO TO 14	ROFF	3836
3		ROFF	3837
3	INSERT STOP CODE AND FLUSH BUFFER.	ROFF	3838
3		ROFF	3839
15	(PAGES=IPAGES+1	ROFF	3840
	DO 16 I=1,5	ROFF	3841
16	CALL PACK (MFC)	ROFF	3842
	CALL PACK (MFC)	ROFF	3843
3		ROFF	3844
	BUFFER OUT (3,1) (LWD(1),LWD(LMP))	ROFF	3845
	IF (UNIT(9)) 17,17,17	ROFF	3846
3		ROFF	3847
17	NMTST=NMTST+5*LMP	ROFF	3848
3		ROFF	3849
	LWD(24)=160015001600160016003	ROFF	3850
	LWD(25)=160015001600160016003	ROFF	3851
3		ROFF	3852
	LBY=0	ROFF	3853
	LMP=26	ROFF	3854
	IF (DONE.OR.(NMTST.GT.13312)) GO TO 18	ROFF	3855
3		ROFF	3856
3	NOT YET DONE	ROFF	3857
3		ROFF	3858
	GO TO 5	ROFF	3859
3		ROFF	3860
3	WRITE LAST RECORD.	ROFF	3861
3		ROFF	3862
18	LWD(26)=16541537000000000000B	ROFF	3863
3		ROFF	3864
	BUFFER OUT (3,1) (LWD(1),LWD(26))	ROFF	3865
	IF (UNIT(9)) 19,19,19	ROFF	3866
3		ROFF	3867
19	END FILE 9	ROFF	3868
	NMTST=NMTST+150	ROFF	3869
	CALL DISPLA (19NMTST CHAR WRITTEN =,NMTST)	ROFF	3870
	IF (DONE) GO TO 20	ROFF	3871
3		ROFF	3872
3		ROFF	3873
3	NOT DONE. PREPARE PROLOGUE FOR NEXT MTST CARTRIDGE.	ROFF	3874
		ROFF	3875
	NMTST=NMTST+NMTST	ROFF	3876
	NMTST=0	ROFF	3877
	NCART=NCART+1	ROFF	3878
	I4=MOD (NCART,10)+1	ROFF	3879
	I3=MOD (NCART/10,10)+1	ROFF	3880
	I2=MOD (NCART/100,10)+1	ROFF	3881
	I1=MOD (NCART/1000,10)+1	ROFF	3882
	LWD(24)=160015001600160016003	ROFF	3883
	LBY=0	ROFF	3884
	LMP=25	ROFF	3885

AFWL-TR-72-139

SUBROUTINE WRT9209

CALL PACK (KDIG(I1))  
CALL PACK (KDIG(I2))  
CALL PACK (KDIG(I3))  
CALL PACK (KDIG(I4))  
CALL PACK (MSTX)

RETURN

DONE. WRITE SECOND ENDFILE, REWIND TAPE9 AND QUIT.

END FILE 9  
REWIND 9

RETURN

ENTRY FIN  
DONE=.TRUE.  
IPAGES=IPAGES-1  
GO TO 14

END

ROFF	3886
ROFF	3887
ROFF	3888
ROFF	3889
ROFF	3890
ROFF	3891
ROFF	3892
ROFF	3893
ROFF	3894
ROFF	3895
ROFF	3896
ROFF	3897
ROFF	3898
ROFF	3899
ROFF	3900
ROFF	3901
ROFF	3902
ROFF	3903
ROFF	3904
ROFF	3905
ROFF	3906

IDENT PACK  
PROGRAM LENGTH

ROFF 3907

BLOCKS

PROGRAM\* LOCAL  
// COMMON  
BLK1 COMMON

ENTRY POINTS

000001 PACK

*	ENTRY PACK		ROFF	3908
*			ROFF	3909
*	SUBROUTINE PACK (WORD)		ROFF	3910
*	PACKS 5 12-BIT BYTES IN LWD(LWP).		ROFF	3911
*	ROUTINE BY HARRY M. MURPHY, 1 FEBRUARY 1972.		ROFF	3912
*	REVISED FOR TN BY LT. CLIFFORD E. RHOADES, JR.		ROFF	3913
*	26 FEBRUARY 1972		ROFF	3914
*			ROFF	3915
	JSE //		ROFF	3916
LWD	3SS 3710		ROFF	3917
	JSE /BLK1/		ROFF	3918
LBV	3SS 1		ROFF	3919
LWP	3SS 1		ROFF	3920
	JSE 0		ROFF	3921
*			ROFF	3922
	VFD 42/4LPACK, 18/1		ROFF	3923
	PS		ROFF	3924
	SA1 X1	.X1 = WORD.	ROFF	3925
	MX0 48	.FORM 48-BIT MASK IN UPPER X0.	ROFF	3926
	SA2 LBV	.X2 = LBV, THE BYTE COUNT.	ROFF	3927
	SB3 5	.B3 = 5.	ROFF	3928
	BX1 -X0*X1	.MASK OUT POSSIBLE HIGH-ORDER BITS IN X1.	ROFF	3929
	SB2 X2	.B2 = LBV.	ROFF	3930
	SB6 LWD-1	.B6 = ADDRESS OF LWD(0).	ROFF	3931
	SB7 1	.B7 = 1.	ROFF	3932
	LT B2,B3,50	.IF LESS THAN 5 BYTES SKIP ON.	ROFF	3933
	SA3 A2+B7	.OTHERWISE, GET LWP IN X3.	ROFF	3934
	SX6 X3+B7	.INCREMENT LWP IN X6.	ROFF	3935
	SA6 A3	.RE-STORE INCREMENTED LWP.	ROFF	3936
	SX2 B0	.AND SET LBV TO ZERO.	ROFF	3937
GO	SA3 A2+B7	.X3 = LWP.	ROFF	3938
	SA4 X3+B6	.X4 = LWD(LWP).	ROFF	3939
	LX4 12	.LEFT SHIFT X4 1 BYTE.	ROFF	3940
	SX7 X2+B7	.INCREMENT BYTE COUNT.	ROFF	3941
	BX5 X0*X4	.MASK OUT LOWER 12 BITS OF LWD(LWP).	ROFF	3942
	SA7 A2	.STORE CURRENT BYTE COUNT.	ROFF	3943
	SX6 X1+X5	.SPLICE IN WORD.	ROFF	3944
	SA6 A4	.STORE UPDATED WORD IN LWD(LWP).	ROFF	3945
	ZR B0,PACK	.AND LOOP TO RETURN.	ROFF	3946
			ROFF	3947

END

UNUSED STORAGE

41 STATEMENTS

5 SYMBOLS

AFWL-TR-72-139

SUBROUTINE QUIT

SUBROUTINE QUIT (IDUM)	ROFF	3948
DIMENSION MES(5)	ROFF	3949
COMMON /TAPE/ ITAPE	ROFF	3950
COMMON /FRAME/ IFRAME	ROFF	3951
COMMON /CARDS/ NC,MI(5)	ROFF	3952
COMMON /PAGES/ IPAGE,ME(5)	ROFF	3953
DATA MES/3JM NUMBER OF FRAMES SHOT BY ROFF/	ROFF	3954
DATA ME(4)/0/	ROFF	3955
DATA MI(4)/0/	ROFF	3956
DATA MES(4)/0/	ROFF	3957
IF(ITAPE.EQ.1) CALL FIN(0)	ROFF	3958
CALL DISPLA(MI,NC)	ROFF	3959
CALL DISPLA(ME,S,IFRAME)	ROFF	3960
CALL DISPLA(ME,IPAGE)	ROFF	3961
RETURN	ROFF	3962
END	ROFF	3963